Personal computer system with protected storage for interface and system utility programs.

Publication number: DE69130042 (T2)
Publication date: 1999-04-15
Inventor(s): ARNOLD LISA RUOTOLO [US]; BEALKOWSKI RICHARD [US]; BLACKLEDGE JOHN WILEY [US]; CRONK DOYLE STANFILL [US]; DAYAN RICHARD ALAN [US]; GEISLER DOUGLAS RICHARD [US]; MITTELSTEDT MATTHEW TODD [US]; PALKA MATTHEW STEPHEN [US]; PAUL JOHN DAVID [US]; SACHSENMAIER ROBERT [US]; SMELTZER KENNETH DONALD [US]; WOYTOVECH PETER ANDREW [US]; ZYVOLOSKI KEVIN MARSHALL [US]
Applicant(s): IBM [US]
Classification:
- international: ***G06F3/06; G06F9/445; G06F13/10;*** *G06F11/08; G06F11/22;* **G06F3/06; G06F9/445; G06F13/10;** G06F11/08; G06F11/22; (IPC1-7): G06F9/445
- European: G06F9/445B6
Application number: DE19916030042T 19910603
Priority number(s): US19900557334 19900723

Also published as:
EP0468625 (A2)
EP0468625 (A3)
EP0468625 (B1)
US5128995 (A)
PT98412 (A)
more >>

Abstract not available for DE 69130042 (T2)
Abstract of corresponding document: **EP 0468625 (A2)**

A personal computer system according to the present invention comprises a system processor, a random access memory, a read only memory, and at least one direct access storage device. A direct access storage device controller coupled between the system processor and direct access storage device includes a protection mechanism for protecting a region of the storage device. The protected region of the storage device includes a master boot record, a BIOS image and a system reference diskette image. The BIOS image includes a section known as Power on Self Test (POST). POST is used to test and initialise a system. Upon detecting any configuration error, system utilities from the system reference diskette image, such as set configuration programs, diagnostic programs and utility programs can be automatically activated from the direct access storage device.

(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Übersetzung der europäischen Patentschrift**

(87) **EP 0 468 625 B 1**

(10) **DE 691 30 042 T 2**

(51) Int. Cl.[6]: **G 06 F 9/445**

(21) Deutsches Aktenzeichen: 691 30 042.9
(86) Europäisches Aktenzeichen: 91 305 020.9
(86) Europäischer Anmeldetag: 3. 6. 91
(87) Erstveröffentlichung durch das EPA: 29. 1. 92
(87) Veröffentlichungstag der Patenterteilung beim EPA: 26. 8. 98
(47) Veröffentlichungstag im Patentblatt: 15. 4. 99

(30) Unionspriorität:
557334 23. 07. 90 US

(73) Patentinhaber:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Teufel, F., Dipl.-Phys., Pat.-Anw., 70569 Stuttgart

(84) Benannte Vertragstaaten:
AT, BE, CH, DE, ES, FR, GB, IT, LI, SE

(72) Erfinder:
Arnold, Lisa Ruotolo, Boynton Beach, FL 33436, US; Bealkowski, Richard, Delray Beach, FL 33444-1033, US; Blackledge, John Wiley, Jr., Boca Raton, FL 33487, US; Cronk, Doyle Stanfill, Delray Beach,FL 33484, US; Dayan, Richard Alan, Boca Raton, FL 33487, US; Geisler, Douglas Richard, Boca Raton, FL 33434, US; Mittelstedt, Matthew Todd, Delray Beach, FL 33446, US; Palka, Matthew Stephen, Jr., Raleigh, NC 27614, US; Paul, John David, Boynton Beach, FL 33426, US; Sachsenmaier, Robert, Boca Raton, FL 33487, US; Smeltzer, Kenneth Donald, Delray Beach, FL 33444, US; Woytovech, Peter Andrew, Boynton Beach, FL 33437-2021, US; Zyvoloski, Kevin Marshall, Raleigh, NC 27614, US

(54) Personalrechnersystem mit geschütztem Speicher für die Schnittstelle und System-Utility-Programme

DE 691 30 042 T 2

DE 691 30 042 T 2

B E S C H R E I B U N G

Personalcomputersystem mit geschütztem Speicher für die Schnittstelle und System-Utility-Programme

Die vorliegende Erfindung betrifft Personalcomputersysteme und im besonderen Mittel zum Schützen und Speichern von Systemdienstprogrammen in einer Direktzugriffs-Speichervorrichtung des Personalcomputersystems.

Querverweis auf einschlägige Patentanmeldungen

Die vorliegende Patentanmeldung gehört zu einer Gruppe von gleichzeitig anhängigen Anmeldungen, die das gleiche übergeordnete Personalcomputersystem betreffen, die jedoch individuell unterschiedliche erfindungsgemäße Konzepte beanspruchen, die in solchen Personalcomputersystemen ausgeführt sind. Diese zugehörigen Patentanmeldungen gelten als Stand der Technik gemäß Artikel 54(3) EPÜ und sind insbesondere wie folgt identifiziert:

Europäische Patentanmeldung Nr. 90307295.7, veröffentlicht unter EP-A-0419904

Europäische Patentanmeldung Nr. 90307297.3, veröffentlicht unter EP-A-0417888

Europäische Patentanmeldung Nr. 90307301.3, veröffentlicht unter EP-A-0417889

Europäische Patentanmeldung Nr. 90307307.0, veröffentlicht unter EP-A-0419005

Zugrundeliegender allgemeiner Stand der Technik

Personalcomputersysteme im allgemeinen und IBM-Personalcomputer im besonderen haben weitverbreitete Anwendung gefunden zum Bereitstellen von Rechnerleistung auf vielen Sektoren der heutigen modernen Gesellschaft. Personalcomputersysteme können üblicherweise definiert werden als Tischgeräte, Bodenstandgeräte oder tragbare Mikrocomputer, die aus einer Systemeinheit mit einem einzigen Systemprozessor, einem Anzeigebildschirm (Monitor), einer Tastatur und einem oder mehreren Diskettenlaufwerken, einem Festplattenspeicher und wahlweise einem Drucker bestehen. Eines der unterscheidenden Merkmale dieser Systeme ist die Anwendung eines Motherboards (Hauptplatine) d.i. einer systemintegrierten Schaltung, um diese Komponenten elektrisch zu verschalten. Diese Systeme sind in erster Linie konstruiert, um für einen einzelnen Anwender unabhängig Rechnerleistung bereitzustellen, und sind preiswert, bestimmt zum Kauf durch Einzelpersonen oder kleinere Geschäftsbetriebe. Beispiele für solche Personalcomputersysteme sind der IBM PERSONAL COMPUTER AT und das IBM PERSONAL SYSTEM/2, Baumuster 25, 30, 50, 50Z, 55SX, 60, 65SX, 70 und 80.

Diese Systeme lassen sich in zwei große Familien einteilen. Die erste Familie, üblicherweise bezeichnet als Family I Modelle, benutzt eine Bus-Architektur, wie beispielhaft im IBM PERSONAL COMPUTER AT und in anderen "IBM-kompatiblen" Maschinen. Die zweite Familie, bezeichnet als Family II Modelle, benutzt die IBM Micro-Channel-Busarchitektur, wie beispielhaft in den IBM PERSONAL SYSTEM/2 Modellen 50 bis 80.

Beginnend mit dem frühesten Personalcomputersystem der Family I Modelle, wie dem IBM Personal Computer, wurde erkannt, daß Software-Kompatibilität von höchster Bedeutung war. Um dieses

Ziel zu erreichen wurde eine Isolierschicht, bestehend aus einem systemresidenten Code, auch als "Firmware" bezeichnet, zwischen der Hardware und der Software eingerichtet. Diese Firmware sah eine Betriebsschnittstelle zwischen einem Applikationsprogramm/Betriebssystem eines Anwenders und der Vorrichtung vor, um den Anwender von Überlegungen im Hinblick auf die Merkmale der Hardware-Vorrichtungen zu befreien. Schließlich entwickelte sich der Code zu einem BASIC-Eingabe/Ausgabe-System (BIOS), damit es möglich wurde, neue Vorrichtungen an das System anzuschließen und dabei während Applikationsprogramm gegen die Besonderheiten der Hardware abzuschotten. Die Bedeutung des BIOS war sofort augenfällig, weil es die Vorrichtungstreiber davon befreite, von spezifischen Hardwarecharakteristiken abhängig zu sein und dabei den Vorrichtungstreiber mit einer unmittelbaren Schnittstelle zur Vorrichtung auszurüsten. Da das BIOS ein integraler Teil des Systems war und die Datenbewegungen zu und von dem Systemprozessor steuerte, war es in der systemintegrierten Schaltung resident und wurde dem Anwender in einem Festspeicher (ROM) geliefert. Zum Beispiel besetzte das BIOS im Original-IBM-Personalcomputer resident 8 K ROM auf der Hauptplatine.

Als dann neue Modelle der Personalcomputerfamilie eingeführt wurden, mußte das BIOS aktualisiert und erweitert werden, um neue Hardware und E/A-Vorrichtungen einzuschließen. Wie zu erwarten war, nahm das BIOS an Speichergröße zu. Zum Beispiel erforderte das BIOS mit der Einführung des IBM PERSONAL COMPUTER AT eine Vergrößerung auf 32 kBytes ROM.

Heute, mit der Entwicklung der neuen Technologie, wachsen die Personalcomputersysteme der Family II Modelle noch komplexer an und stehen den Anwendern auch häufiger zur Verfügung. Da sich die Technologie rapide ändert und neue E/A-Vorrichtungen zum Personalcomputersystem hinzugefügt werden, wurde die

BIOS-Änderung ein signifikantes Problem im Entwicklungszyklus der Personalcomputersysteme.

So wurde zum Beispiel mit der Einführung des IM Personal System/2 mit Micro-Channel-Architektur ein signifikant neues BIOS, bekannt als Advanced BIOS oder ABIOS, entwickelt. Um aber die Software-Kompatibilität zu erhalten, mußte das BIOS der Modelle der Family I in die Modelle der Family II übernommen werden. Das Family I BIOS wurde bekannt als Compatibility-BIOS oder CBIOS. Aber wie bereits im Hinblick auf den IBM PERSONAL COMPUTER AT erklärt, waren nur 32 kBytes ROM auf der Hauptplatine resident. Glücklicherweise konnte das System auf 96 kBytes ROM erweitert werden. Leider stellte sich heraus, daß das aufgrund technischer Zwänge die maximal mögliche Kapazität für das BIOS war. Glücklicherweise konnten sogar mit Zusatz des ABIOS, das ABIOS und CBIOS in 96 K ROM gequetscht werden. Es blieb jedoch nur ein kleiner Prozentsatz des 96K ROM-Platzes für Erweiterungen frei. Bei zusätzlicher Aufnahme künftiger E/A-Vorrichtungen werden CBIOS und ABIOS schließlich keinen Platz mehr auf dem ROM haben. Somit ist es nicht möglich, neue E/A-Technologien einfach in CBIOS und ABIOS zu integrieren.

Wegen dieser Probleme plus dem Wunsch, Änderungen im Family II BIOS möglichst spät in den Entwicklungszyklus einzubringen, mußten Teile des BIOS aus dem ROM ausgelagert werden. Das wurde erreicht durch Speichern von Teilen des BIOS auf einer Massenspeichervorrichtung wie z.B. einer Festplatte. Da eine solche Platte sowohl Schreibe- als auch Lese-Fähigkeit vorsieht, wurde es möglich, den aktuellen BIOS-Code auf der Platte zu verändern. Zwar bot die Platte eine schnelle und wirksame Methode zum Abspeichern des BIOS-Code, jedoch vergrößerte sich auch die Wahrscheinlichkeit, daß der BIOS-Code verfälscht wird. Da das BIOS ein integraler Teil des Betriebssystems ist, könnte ein verfälschtes BIOS zu vernich-

tenden Ergebnissen führen und in manchen Fällen zum Totalabsturz, ja zur Betriebsunfähigkeit des Systems führen. So war es ganz folgerichtig, daß Mittel zum Verhüten einer unzulässigen Veränderung des BIOS-Code auf der Festplatte höchst erwünscht waren, das war der Gegenstand der Europäischen Patentanmeldung 90307301.3.

Zusätzlich zum Abspeichern des BIOS auf einer Massenspeichervorrichtung wurde auch das Abspeichern von Systemdienstprogrammen, die in der Regel auf einer systembezüglichen Diskette enthalten sind, sehr erwünscht. Der Ausschluß der Systemdiskette verringert nicht nur den Preis des Systems, sondern sieht auch eine anwenderfreundlichere Umgebung vor.

Hier sollte kurz der Zweck der Systemdienstprogramme erläutert werden, die vorher auf der Referenzdiskette abgespeichert waren. Mit der Einführung des PS/2 Micro Channel System von IBM kam der Abbau von Schaltern und Drahtbrücken von den E/A-Adapterkarten und der Hauptplatine. Die Micro Channel Architektur sah programmierbare Register zu deren Ersatz vor. Dienstprogramme zum Konfigurieren dieser programmierbaren Register d.i. programmierbaren Optionswahlregister (Programmable Option Select - POS) wurden erforderlich. Zusätzlich wurden noch andere Dienstprogramme zur Verbesserung der System-Anwendbarkeitsmerkmale zusammen mit einem Systemdiagnoseprogramm auf dieser Systemdiskette versandt.

Vor der Erstanwendung erforderte jedes Micro Channel System die Initialisierung seiner POS-Register. Zum Beispiel, wenn das System mit einer neuen E/A-Karte oder einem geänderten Einbauplatz für eine E/A-Karte gebootet wird, wird ein Konfigurationsfehler generiert und das System-Urladeverfahren bleibt stehen. Der Anwender wird dann aufgefordert, die Systemdiskette zu laden und die Taste F1 zu drücken. Dann kann von der Systemdiskette ein "Set Configuration Utility"

(Konfigurations-Dienstprogramm) gebootet werden, um das System zu konfigurieren. Das Set Configuration Utility Dienstprogramm führt dann den Anwender zur gewünschten Aktion. Wenn die geeigneten Deskriptordateien auf der E/A-Karte geladen sind, generiert das Set Configuration Utility Programm die richtige POS oder Konfigurationsdaten in einem nichtflüchtigen Speicher. Die Deskriptordatei enthält Konfigurationsinformationen, die die Karte über eine Schnittstelle mit einem System zusammenbinden.

Obwohl dieses Verfahren verhältnismäßig einfach durchzuführen ist, muß die Systemdiskette bereitliegen oder leicht zugänglich in der Nähe aufbewahrt werden. Es ist schon vorgekommen, daß nach einiger Zeit die Systemdiskette verlegt wurde. Daher ist es sehr erwünscht, eine Kopie der Systemdiskette zusammen mit dem BIOS auf der Massenspeichervorrichtung abzuspeichern, um die Einsatzfähigkeit des Systems zu verbessern.

Gemäß der vorliegenden Erfindung ist vorgesehen ein Personalrechnersystem mit einem Systemprozessor zum Abarbeiten eines Betriebssystems, einem Festwertspeicher, einem Speicher mit wahlfreiem Zugriff und Peripheriegeräten einschließlich wenigstens einer Direktzugriffsspeichervorrichtung, wobei das System ein Schnittstellenprogramm zur Steuerung der Datenbewegung in den bzw. aus dem Prozessor aufweist, dadurch gekennzeichnet, daß die Direktzugriffsspeichervorrichtung ein Schutzmittel aufweist zum Schützen eines Bereichs der Direktzugriffsspeichervorrichtung, in der ein Teil des Schnittstellenprogramms gespeichert ist, das Schutzmittel den Zugriff auf den geschützten Bereich als Reaktion auf ein Rückstellsignal zuläßt, dieser Teil des Schnittstellenprogramms betriebsfähig ist, wenn er in den Speicher mit wahlfreiem Zugriff geladen ist, um das Betriebssystem zu booten und das Schutzmittel zu aktivieren, um den Zugriff auf den Schutzbereich der Direktzugriffsspeichervorrichtung während des Abar-

beitens des Betriebssystems zu verhindern, wobei der Schutzbereich der Direktzugriffsspeichervorrichtung ferner Systemdienstprogramme beinhaltet, die abgearbeitet werden, sobald beim Laden des Betriebsystems ein Fehlerzustand erkannt wird.

Gemäß einer Ausführungsform der vorliegenden Erfindung umfaßt das Personalcomputersystem einen Systemprozessor, einen Speicher mit wahlfreiem Zugriff, einen Festwertspeicher und mindestens eine Speichervorrichtung mit direktem Zugriff. Ein Diektzugriffs-Speichervorrichtungs-Controller kann zwischen den Systemprozessor und die Direktzugriffs-Speichervorrichtung geschaltet sein und enthält ein Mittel zum Schützen eines Bereichs der Speichervorrichtung. Der geschützte Bereich der Speichervorrichtung kann eine Master-Startroutine, ein BIOS-Bild und das Systemdiskettenbild enthalten. Das BIOS-Bild beinhaltet einen Abschnitt, bekannt als Power-On-Self-Test (POST - Selbsttest beim Einschalten). Der POST wird benutzt, um ein System zu testen und zu initialisieren. Beim Erfassen eines Konfigurationsfehlers können Systemdienstprogramme vom Systemdiskettenbild, wie Einstell-Konfigurationsprogramme, diagnostische Programme und Dienstprogramme, automatisch aktiviert werden.

Insbesondere erlaubt, als Reaktion auf ein Rückstellsignal zum Booten des Systems, das Schutzmittel den Zugriff auf den geschützten Bereich, um die Master-Startroutine in den Speicher mit wahlfreiem Zugriff zu laden. Im Betrieb lädt die Master-Startroutine ferner das BIOS-Bild in den Speicher mit wahlfreiem Zugriff. Das BIOS, das jetzt im Speicher mit wahlfreiem Zugriff sitzt, wird abgearbeitet und bootet das Betriebssystem zum Anfahren des Systems, und dann generiert BIOS ein zweites Signal, das das Schutzmittel aktiviert, um den Zugriff auf den Bereich der Platte, der die Master-Startroutine und das BIOS-Bild enthält, zu sperren. Wenn das BIOS (POST) einen Fehler findet, generiert das BIOS ein drittes

Signal, um das Schutzmittel auszuschalten, und versucht dann eine Systemreferenzdiskette, die in einem ladbaren Diskettenlaufwerk steht, zu booten. Wenn keine Systemreferenzdisektte vorhanden ist, dann bootet BIOS die Systemdienstprogramme in den Systempartitionsbereich.

Im besonderen beinhaltet der Festwertspeicher einen ersten Teil des BIOS. Dieser erste Teil des BIOS initialisiert den Systemprozessor, die Diektzugriffs-Speichervorrichtung, und stellt das Schutzmittel zum Lesen der Master-Startroutine vom geschützten Bereich oder der Partition auf der Direktzugriffs-Speichervorrichtung in den Speicher mit wahlfreiem Zugriff zurück. Die Master-Startroutine beinhaltet ein Datensegment und ein ausführbares Code-Segment. Das Datensegment beinhaltet Daten, die eine Systemhardware repräsentieren, und eine Systemkonfiguration, die von der Master-Startroutine unterstützt wird. Der erste BIOS-Teil bestätigt, daß die Master-Startroutine kompatibel mit der Systemhardware ist durch Überprüfen der Daten aus dem Datensegment, ob die Masterstartroutine mit den Daten im ersten BIOS-Teil übereinstimmt, der den Systemprozessor, die System-Hauptplatine und die Hauptplatinen-E/A-Konfiguration repräsentieren.

Wenn die Master-Startroutine kompatibel mit der System-Hardware ist, führt der erste BIOS-Teil den Zeiger des System-Prozessors zur Abarbeitung des ausführbaren Codesegments der Master-Startroutine. Das ausführbare Codesegment bestätigt, daß sich die Systemkonfiguration nicht verändert hat, und lädt den restlichen BIOS-Teil aus der Direktzugriffs-Speichervorrichtung in den Speicher mit wahlfreiem Zugriff. Dann überprüft das ausführbare Codesegment die Gültigkeit des restlichen BIOS-Teils, führt den Zeiger des Systemprozessors zum Ausführen des BIOS, jetzt im Speicher mit wahlfreiem Zugriff. Das im Speicher mit wahlfreiem Zugriff ablaufende BIOS generiert das zweite Signal zum Schutz der Plattenpartition,

in der das restliche BIOS steht, und bootet dann das Betriebssystem zum Anfahren des Personalcomputersystems. Die Partition, die das restliche BIOS enthält, ist geschützt, um den Zugriff zum BIOS-Code auf der Platte zu verhindern und die Integrität des BIOS-Code zu schützen.

Wenn dann aber entweder ein Fehler oder eine vom Anwender aufgerufene Diagnose-Boot-Schlüsselsequenz vom BIOS vor dem Booten des Betriebssystems gefunden wird, wird das Systemreferenzdiskettenbild, falls vorhanden, von der Systempartition gebootet. Zusätzlich, wenn eine Systemreferenzdiskette in Diskettenlaufwerk A gefunden wird, bekommt die Systemreferenzdiskette den Vorrang über das Bild in der Systempartition und wird statt dessen gebootet. In solchen Situationen stellt BIOS sicher, daß die Schutzmittel deaktiviert sind bevor der Urlader die Steuerung über die Startroutine abgibt. Somit ist das Schutzmittel zum Verhindern des Zugriffs auf den Plattenbereich, in dem die Master-Startroutine, das BIOS-Bild und das Systemreferenzdiskettenbild enthalten ist, nicht aktiv. Dann lädt BIOS das Systemreferenzdiskettenbild bzw. die Systemreferenzdiskette zusammen mit dem Bereich auf der Platte, der offen zum Zugriff durch die Software ist.

In den Zeichnungen:

Fig. 1 zeigt eine Schnittansicht eines Personalcomputersystems, das die erfindungsgemäße Ausführungsform darstellt, und eine Systemhauptplatine zeigt, die mit einer Vielzahl von Direktzugriffsspeichervorrichtungen verbunden ist;

Fig. 2 zeigt ein System-Blockschaltbild für das Personalcomputersystem der Fig. 1;

Fig. 3 ist ein Speicherabbild für das ROM BIOS, eingeschlossen in der Hauptplatine;

Fig. 4 ist ein Flußdiagramm, das den Gesamtprozeß zum Laden eines BIOS-Bilds von einer Speichervorrichtung mit direktem Zugriff lädt;

Fig. 5 zeigt das Satzformat der Master-Startroutine;

Fig. 6A ist ein Flußdiagramm, das die Operation der IBL-Routine zeigt;

Fig. 6B ist ein Flußdiagramm, das die Schritte zum Laden eines BIOS-Bildes von einer Festplatte zeigt;

Fig. 6C ist ein Flußdiagramm, das die Schritte zum Laden eines BIOS-Bildes von einer Diskette zeigt;

Fig. 6D ist ein Flußdiagramm, das in näheren Einzelheiten die Prüfung der Kompatilbilität zwischen Master-Startroutine und Hauptplatine/Prozessor zeigt;

Fig. 7 ist ein detailliertes Flußdiagramm, das den Betrieb des ausführbaren Code-Segments der Master-Startroutine zeigt;

Fig. 8 ist ein Blockdiagramm, das den Controller der Speichervorrichtung mit direktem Zugriff zeigt;

Fig. 9 ist ein Flußdiagramm, das den Betrieb eines Platten-Controller zum Schutz des auf einer Platte abgespeicherten IBL-Mediums zeigt.

Fig. 10 ist ein Flußdiagramm, das ein Verfahren zum Schutz des BIOS-Bilds zeigt;

Fig. 11 ist ein Flußdiagramm, das den Prozeß zur Entscheidungsfindung zeigt, wann das Systemreferenzdiskettenbild von einer Direktzugriffs-Speichervorrichtung geladen werden soll;

Fig. 12 ist ein Flußdiagramm, das zeigt, wie der Urlader das richtige Medium einschließlich des Systemreferenzdiskettenbilds von einer Direktzugriffsspeicher-Vorrichtung lädt; und

Fig. 13 ist ein Flußdiagramm, das die Änderungen am BIOS zeigt, um die Behandlung der Systempartition als aktive Partition auf einer Festplatte einschalten zu können.

Nehmen wir jetzt Bezug auf die Zeichnungen und insbesondere auf Fig. 1; dort ist eine Schnittansicht eines Personalcomputersystems 10 gezeigt, mit einer Vielzahl von DASD (Direct Access Storage Devices - Direktzugriffs-Speichervorrichtungen) 12 - 16, die über eine Vielzahl E/A-Schlitze 18 an eine System- oder Hauptplatine 24 angeschlossen sind. Eine Stromzuführung 22 versorgt das System 10 auf wohlbekannte Art mit elektrischem Strom. Die Hauptplatine 24 beinhaltet einen Systemprozessor, der unter der Steuerung der Rechneranweisungen arbeitet, um Informationen einzugeben, zu verarbeiten und auszugeben.

In der Praxis ist das Personalcomputersystem 10 in erster Linie so konstruiert, daß es unabhängige Rechnerleistung an eine kleine Anwendergruppe oder an einen einzigen Anwender gibt, und ist preiswert gehalten zum Ankauf durch Einzelpersonen oder kleine Geschäfte. Im Betrieb arbeitet der Systemprozessor unter einem Betriebssystem wie z.B. unter dem Betriebssystem OS/2 der IBM oder DOS. Dieser Betriebssystemtyp beinhaltet eine BIOS-Schnittstelle zwischen dem DASD 12 - 16 und dem Betriebssystem. Ein Teil des BIOS, aufgeteilt in Funktionsmodule, ist in einem ROM auf der Hauptplatine 24 abgespeichert und wird nachstehend als ROM-BIOS bezeichnet.

BIOS sieht vor eine Schnittstelle zwischen der Hardware und der Betriebssystem-Software, damit der Programmierer bzw. der Anwender seine Maschine programmieren kann, ohne eine tiefere Kenntnis vom Betrieb einer bestimmten Vorrichtung zu haben. Zum Beispiel ermöglicht es ein BIOS-Diskettenmodul einem Programmierer, das Diskettenlaufwerk ohne nähere Kenntnis von der Diskettenlaufwerk-Hardware zu programmieren. Somit kann eine Anzahl Diskettenlaufwerke im System benutzt werden, die von unterschiedlichen Herstellern konstruiert und gefertigt werden. Das senkt nicht nur die Kosten des Systems 10, sondern läßt auch dem Anwender die Wahl zwischen verschiedenen Diskettenlaufwerken.

Vor Inbezugsetzen der obigen Struktur mit der vorliegenden Erfindung sollte eine kurze Zusammenfassung über den allgemeinen Betrieb des Personalcomputersystems 10 gegeben werden. Nehmen wir Bezug auf Fig. 2; dort wird ein Blockdiagramm des Personalcomputersystems 10 gezeigt. Fig. 2 illustriert Komponenten der Hauptplatine 24 und die Anschlüsse der Hauptplatine 24 an die E/A-Schlitze 18 und weiterer Hardware des Personalcomputersystems. Auf der Platine 24 findet sich der Systemprozessor 26, bestehend aus einem Mikroprozessor, der über einen örtlichen Bus 28 mit einem Speicher-Controller 30 verbunden ist, der weiter an einen Speicher mit wahlfreiem Zugriff (RAM) 32 angeschlossen ist. Jeder geeignete Mikroprozessor kann verwendet werden, so ist z.B. ein geeigneter Mikroprozessor der 80386 von Intel.

Der Systemprozessor könnte auch ein Intel 80286 oder 80486 Mikroprozessor sein.

Im Zugriffsbereich des Prozessors liegt eine Hauptplatinen-Kennung (Planar-ID). Die Planar-ID ist unverwechselbar für die Hauptplatine und identifiziert die benutzte Hauptplatine. Zum Beispiel kann die Planar-ID festverdrahtet sein, um durch

einen E/A-Port des Systemprozessors 26 oder durch Schalten von Schaltern gelesen zu werden. Zusätzlich kann ein anderer E/A-Port des Systemprozessors 26 benutzt werden, um unter Verwendung der Hauptplatinen-Logikschaltung ein Rückstellsignal an den Platten-Controller zu generieren. Zum Beispiel kann das Rückstellsignal dadurch eingeleitet werden, daß die Software den E/A-Port anspricht und die Hauptplatinenlogik aktiviert, um das Rückstellsignal zu generieren.

Der örtliche Bus 28 ist ferner über einen Bus-Controller 34 an den Festwertspeicher (ROM) 36 auf der Hauptplatine 24 angeschlossen. Ein zusätzlicher nichtflüchtiger Speicher (NVRAM) 58 ist über eine serielle/parallele Port-Schnittstelle 40, die ferner am Bus-Controller 34 liegt, an den Mikroprozessor 26 angeschlossen. Der nichtflüchtige Speicher kann ein batteriegestützter CMOS sein, um Informationen zu bewahren, wenn immer der Strom vom System abgeschaltet wird. Da der ROM im allgemeinen auf der Hauptplatine resident ist, werden im ROM abgespeicherte Modellwerte und Untermodellwerte benutzt, um den Systemprozessor und die Systemhauptplatinen-E/A-Konfiguration zu identifizieren. Somit werden diese Werte den Prozessor und die Hauptplatinen-E/A-Konfiguration physikalisch identifizieren.

Der NVRAM wird benutzt zum Abspeichern von Systemkonfigurationsdaten. Das heißt, der NVRAM enthält Werte die die vorliegende Konfiguration des Systems beschreiben. Zum Beispiel enthält der NVRAM Informationen, die die Kapazität einer Festplatte oder Diskette, den Anzeigetyp, die Größe des Speichers, Zeit, Datum usw. beschreiben. Zusätzlich werden die im ROM gespeicherten Modell- und Untermodellwerte in den NVRAM kopiert, wenn immer ein besonderes Konfigurationsprogramm, wie z.B. SET Configuration, gefahren wird. Der Zweck des Programms SET Configuration ist, Werte zur Charakterisierung der Konfiguration des Systems im NVRAM zu speichern. So sind

für ein ordentlich konfiguriertes System die Modell- und Untermodellwerte im NVRAM gleich den Modell- bzw. Untermodellwerten, die im ROM gespeichert sind. Wenn diese Werte nicht gleich sind, heißt das, daß die Konfiguration des Systems verändert wurde. Hier nehmen wir Bezug auf Fig. 6D, wo dieses Merkmal anhand des Ladens des BIOS in weiteren Einzelheiten erklärt ist.

Führen wir unsere Diskussion unter Hinweis auf Fig. 2 fort, der Bus-Controller 34 ist ferner über einen E/A-Hauptplatinenbus 43 an E/A-Schlitze 18, die serielle/parallele Schnittstelle 40 und den Peripherie-Controller 42 gekoppelt. Der Peripherie-Controller 42 ist ferner an eine Tastatur 44, eine Maus 46, an einen Diagnoseeinschub 47 und einen Disketten-Controller 64 angeschlossen. Neben dem NVRAM 58 ist die serielle/parallele Schnittstelle 40 ferner an einen seriellen Port 48 und einen parallelen Port 50 gekoppelt, um Informationen an einen Drucker, eine Blattausgabevorrichtung usw. zu schicken. Wie dem Fachmann wohlbekannt, kann der örtliche Bus 28 auch an einen Cache-Controller 52, einen Cache-Speicher 68, einen Co-Prozessor 54 und einen DMA-Controller 56 angeschlossen sein.

Der Systemprozessor 26 steuert seinen internen Betrieb sowie seine Schnittstellenverbindungen mit anderen Elementen des Personalcomputersystems 10. Zum Beispiel wird der Systemprozessor 26 als verbunden mit einer E/A-Karte 60 einer Kleinrechnersystem-Schnittstelle (SCSI) dargestellt, die ferner mit einem DASD, wie z.B. mit einem Festplattenlaufwerk 62 verbunden gezeigt wird. Hier muß darauf hingewiesen werden, daß auch ein anderes als ein SCSI-Festplattenlaufwerk erfindungsgemäß als Festplatte benutzt werden kann. Zusätzlich zur Festplatte 62 kann der Systemprozessor 26 auch über eine Schnittstelle an den Disketten-Controller 64, der ein Diskettenlaufwerk 66 steuert, angeschlossen werden. Für die Termi-

nologie ist darauf hinzuweisen, daß der Ausdruck "Hardfile" das Festplattenlaufwerk 62 bezeichnet, während der Ausdruck "Floppy" auch das Diskettenlaufwerk 66 bezeichnet.

Vor der vorliegenden Erfindung beinhaltete der ROM 36 den gesamten BIOS-Code, der die Schnittstelle zwischen dem Betriebssystem und den Hardware-Peripheriegeräten war. Laut einem Aspekt der vorliegenden Erfindung ist jedoch der ROM 36 so ausgelegt, daß er nur einen Teil des BIOS abspeichert. Wenn dieser Teil durch den Systemprozessor 26 abgearbeitet wird, gibt er entweder von der Festplatte 62 oder von der Diskette 66 einen zweiten Teil, das ist der Rest des BIOS, nachstehend auch als BIOS-Bild bezeichnet, ein. Das BIOS-Bild überschreibt den ersten BIOS-Teil und ist als integrierender Teil des Systems im Hauptspeicher, wie dem RAM 32, resident. Der erste Teil des BIOS (ROM-BIOS), gespeichert im ROM 36, wird allgemein anhand der Fig. 3-4, und im Detail anhand der Fig. 6A-D erklärt. Der zweite Teil des BIOS (BIOS-Bild) wird anhand der Fig. 5 erklärt und das Laden des BIOS-Bildes anhand der Fig. 7. Ein weiterer Vorteil des Ladens eines BIOS-Bilds von einem DASD ist die Fähigkeit, das BIOS direkt in den RAM 32 des Systemprozessors zu laden. Da der Zugriff auf den RAM viel schneller ist als der Zugriff auf den ROM, wird eine signifikante Verbesserung der Arbeitsgeschwindigkeit des Computersystems erzielt. Ein zusätzlicher Vorteil wird gewonnen durch Speichern der System-Dienstprogramme auf dem DASD. Wenn eine Bedingung eintritt, für die der Gebrauch der System-Dienstprogramme erforderlich ist, kann das System-Dienstprogramm automatisch auf dem DASD angesprochen werden.

Die Erklärung geht nun dazu über, den Betrieb des BIOS im ROM 36 und den Vorgang des Ladens des BIOS-Bilds und des Systemreferenzdisketten-Bilds von der Festplatte bzw. von der Diskette zu erklären. Im allgemeinen überprüft ein erstes Programm, z.B. ein ROM-BIOS, das System vorab und lädt eine

BIOS-Masterstartroutine in den RAM. Die Masterstartroutine beinhaltet ein Datensegment mit einer Gültigkeitsinformation und, weil es ein Lademittel ist, ein Codesegment mit einem ausführbaren Code. Der ausführbare Code benutzt die Dateninformation zur Bestätigung der Hardware-Kompatibilität und Systemkonfiguration. Nach dem Testen auf Hardware-Kompatibilität und richtige Systemkonfiguration lädt der ausführbare Code das BIOS-Bild in den RAM und erzeugt ein hauptspeicherresidentes Programm. Das BIOS-Bild folgt auf das ROM-BIOS und lädt das Betriebssystem, um die Maschine anzufahren. Zwecks Klarheit wird das ausführbare Codesegment der Masterstartroutine als MBR-Code und das Datensegment als MBR-Daten bezeichnet.

Nehmen wir Bezug auf Fig. 3, hier wird eine Speicherabbildung dargestellt, die die verschiedenen Code-Module zeigt, aus denen das ROM-BIOS besteht. ROM-BIOS beinhaltet ein Einschalt-Eigentest-(Power On Self Test - POST)-Stufe-I-Modul 70, ein Anfangs-BIOS-Lade-(Initial BIOS Load - IBL)-Routine-Modul 72, ein Disketten-Modul 74, ein Hardfile-Modul 76, ein Videomodul 78, ein Diagnoseanzeigemodul 80, und Hardware-Kompatibilitätsdaten 82. Kurz gesagt, die POST-Stufe I 70 führt die Vorinitialisierung und Tests des Systems durch. Die IBL-Routine 72 bestimmt, ob das BIOS-Bild von der Platte oder von der Diskette geladen werden soll, überprüft die Kompatibilität und lädt die Masterstartroutine. Das Disketten-Modul 74 sieht Eingabe/Ausgabe-Funktionen für ein Diskettenlaufwerk vor. Das Hardfile-Modul 76 steuert die E/A zu einer Festplatte oder dergl. Video-Modul 78 steuert Ausgabefunktionen zu einem Video-E/A-Controller, der ferner mit einem Videobildschirm verbunden ist. Das Diagnostikanzeigemodul 80 übergibt die Steuerung für das System an eine Diagnoseanzeigevorrichtung. Die Hardware-Kompatibilitätsdaten 82 beinhalten Werte wie einen System-Modell- und -Untermodell-Wert, die später anhand Fig. 5 beschrieben werden.

Nehmen wir jetzt Bezug auf Fig. 4; dort wird eine Prozeß-Übersicht zum Laden eines BIOS-Bilds in das System entweder von der Festplatte oder von der Diskette gezeigt. Wenn das System hochgefahren wird, wird der Systemprozessor auf den Eingangspunkt der POST-Stufe I geführt, Schritt 100. Die POST-Stufe I initialisiert das System und testet nur die Systemfunktionen, die zum Laden des BIOS-Bilds vom angewählten DASD erforderlich sind, Schritt 102. Im einzelnen initialisiert POST-Stufe I die Prozessor/Hauptplatinen-Funktionen, die Diagnostik-Anzeige, das Speicheruntersystem, den Interrupt-Controller, die Zeitgeber, das DMA-Untersystem, die Festplatten-BIOS-Routine (Hardfile-Modul 76) und die Disketten-BIOS-Routine (Diskettenmodul 74), falls erforderlich.

Nachdem die POST-Stufe I das System vorinitialisiert hat, führt die POST-Stufe I den Systemprozessor zur Anfangs-BIOS-Laderoutine (IBL), die im Anfangs-BIOS-Lademodul 72 enthalten ist. Die IBL Routine bestimmt zunächst, ob das BIOS-Bild auf der Festplatte gespeichert ist oder von der Diskette geladen werden kann; und zweitens, lädt sie die Masterstartroutine vom angewählten Medium (Platte oder Diskette) in den RAM, Schritt 104. Die Masterstartroutine beinhaltet die MBR-Daten und den MBR-Code. Die MBR-Daten werden für Überprüfungsmaßnahmen benutzt, und der MBR-Code wird zum Laden des BIOS-Bilds abgearbeitet. Eine detaillierte Beschreibung der Operation der IBL-Routine wird anhand Fig. 6A-D gezeigt.

Nehmen wir weiterhin Bezug auf Fig. 4; nachdem die IBL-Routine die Mastersteuerroutine in den RAM geladen hat, wird der Systemprozessor zur Startadresse des MBR-Codes vektorgeführt, Schritt 106. Der MBR-Code führt eine Reihe Gültigkeitstests durch, um die Authentizität des BIOS-Bilds zu bestimmen und um die Konfiguration des Systems zu überprüfen. Zwecks besseren Verständnisses der Operation des MBR-Code

wird die Aufmerksamkeit auf Fig. 7 der Zeichnungen gerichtet, in der der MBR-Code in näheren Einzelheiten beschrieben ist. Auf der Grundlage dieser Gültigkeitstests lädt der MBR-Code das BIOS-Bild in den RAM und übergibt die Steuerung an das eben geladene BIOS-Bild im Hautspeicher, Schritt 108. Im einzelnen wird das BIOS-Bild in den ursprünglich vom ROM-BIOS besetzten Speicherraum geladen. Das heißt, wenn das ROM-BIOS von E0000h bis FFFFFh adressiert ist, dann wird das BIOS-Bild in diesen RAM-Adressenplatz geladen und überschreibt das ROM-BIOS. Dann geht die Steuerung über auf POST-Stufe II, die im neu geladenen BIOS-Bild eingeschlossen ist und verläßt so das ROM-BIOS. POST-Stufe II, jetzt im RAM, initialisiert und testet das restliche System, um den Betriebssystem-Lader zu laden, Schritte 110-114. Bevor POST Stufe II die Steuerung an das Betriebssystem überträgt, setzt die POST-Stufe II einen Schutz, damit der Zugriff auf die Plattenpartition, die das BIOS-Bild enthält, verhindert wird. Wenn jedoch ein Fehler gefunden wird, kann POST Stufe II das Schutzmittel abschalten und die System-Dienstprogramme im Systemreferenzdiskettenbild auf der Platte abrufen. Hier wird Bezug genommen auf Fig. 8-10 für eine detaillierte Diskussion dieses Schutzprozesses. Es wird angemerkt, daß bei einem Warmstart der Prozessor zu Schritt 108 vektorgeführt wird unter Umgehung der Schritte 100-106.

Zwecks Klarheit sollte an diesem Punkt eine Darstellung des Formats der Masterstartroutine gezeigt werden. Unter Bezugnahme auf Fig. 5 wird die Masterstartroutine gezeigt. Die Startroutine beinhaltet das ausführbare Codesegment 120 und die Datensegmente 122-138. Der MBR-Code 120 beinhaltet den DASD-abhängigen Code, der verantwortlich ist für die Überprüfung der Identität des ROM-BIOS durch Prüfung, daß die IBL-Startroutine mit dem System kompatibel ist, Überprüfung der Systemkonfiguration und Laden des BIOS-Bilds vom angewählten DASD (Platte oder Diskette). Die Datensegmente 122-

138 beinhalten Informationen, die benutzt werden, um das Medium zu definieren, die Masterstartroutine zu identifizieren und überprüfen, das BIOS-Bild zu finden und das BIOS-Bild zu laden.

Die Masterstartroutine wird identifiziert durch eine Startroutinesignatur 122. Die Startroutinesignatur 122 kann z.B. ein unverwechselbares Bitmuster, wie eine Buchstabenkette "ABC" in den ersten drei Bytes des Datensatzes sein. Die Integrität der Masterstartroutine wird getestet durch einen Prüfsummenwert 132, der beim Laden der Startroutine mit einer berechneten Prüfsumme verglichen wird. Die Datensegmente beinhalten ferner mindestens einen kompatiblen Hauptplatinen-ID-Wert 134, kompatible Modell- und Untermodellwerte 136. Der Hauptplatinen-ID-Wert der Masterstartroutine definiert, für welche Hauptplatine die Masterstartroutine gültig ist. Auf ähnliche Weise definieren die Modell- und Untermodellwerte der Masterstartroutine den Prozessor und die Hauptplatinen-E/A-Konfiguration, für die die Masterstartroutine gilt. Bemerkt wird, daß die Startroutinesignatur und Prüfsumme eine gültige Masterstartroutine identifizieren, während die Startroutine-Hauptplatinen-ID, Startroutinemodell- und Startroutineuntermodellvergleiche benutzt werden, um eine Startroutine zu identifizieren, die mit dem System kompatibel ist, und um zu bestimmen, ob die Systemkonfiguration gültig ist. Ein anderer Wert, Startroutinemuster 124 wird benutzt, um die Gültigkeit des ROM-BIOS zu bestimmen. Das Startroutinemuster 124 wird verglichen mit einem entsprechenden Musterwert, der im ROM gespeichert ist. Wenn der Wert mit diesem übereinstimmt, zeigt er an, daß ein gültiger ROM-BIOS das Laden eines BIOS-Bildes vom angewählten Medium eingeleitet hat.

Die nachstehende Beschreibung beschreibt weiter detailliert jeden der Werte in der Masterstartroutine und deren Funktionen:

MBR-Kennung 122: Die ersten drei Bytes der IBL-Startroutine können aus Zeichen bestehen wie z.B. "ABC". Diese Signatur wird zur Identifizierung eine Startroutine benutzt.

MBR-Codesegment 120: Dieser Code überprüft die Kompatibilität der Startroutine mit der Hauptplatine und dem Prozessor durch Vergleichen entsprechender Hauptplatinen-ID- und Modell/Untermodell-Werte. Wenn diese Werte übereinstimmen, lädt er das BIOS-Bild vom angewählten Medium in den System-RAM. Wenn die Systembild-(im Speicher geladenes BIOS-Bild)-Prüfsumme gültig ist und kein Medium-Ladefehler auftritt, überträgt der MBR-Code die Steuerung auf die POST-Stufe II-Routine des Systembilds.

MBR-Muster 124: Das erste Feld des IBL-Startroutine-Datums-Segments enthält ein Muster, wie z.B. eine Zeichenkette "ROM-BIOS 1990". Diese Kette wird benutzt zur Gültigkeitsüberprüfung des ROM-BIOS durch Vergleich des Urlademusterwerts mit dem entsprechenden Wert, der im ROM gespeichert ist (ROM-Muster).

MBR-Versionsdatum 126: Die Masterstartroutine enthält ein Versionsdatum zur Verwendung durch ein Aktiualisierungs-Dienstprogramm.

Systempartitionszeiger 128: Das Datensegment enthält einen Medium-Zeiger auf den Anfang des Mediumsystempartitionsbereichs zur Verwendung durch die POST Stufe II. Auf einer IBL-Diskette steht der Zeiger im Spurkopfsektor-Format, auf der Platte steht der Zeiger im Relativblockadressen-(RBA)-Format.

Systempartitionstyp 130: Der Systempartitionstyp zeigt die Struktur der Mediumsystempartition. Es gibt drei Arten der

Systempartitionsstrukturen - voll, minimal und nicht vorhanden. Die volle Systempartition enthält das Setup-Dienstprogramm und die Diagnostik zusätzlich zum BIOS-Bild und der Masterstartroutine. Die minimale Systempartition enthält nur das BIOS-Bild und die Masterstartroutine. Sie kann auftreten, wenn ein System keinen Zugriff auf eine Hardfile mit einem IBL-Bild hat, unter diesen Umständen zeigt der Systempartitionstyp 'nicht vorhanden' an. In diesem Fall kommt IBL von der Diskette. Diese drei Systempartitionstypen ermöglichen Flexibilität, wieviel Platz die Systempartition auf dem Medium beansprucht.

Prüfsummenwert 132: Der Prüfsummenwert des Datensegments wird initialisiert, um eine gültige Prüfsumme für den Datensatzlängenwert (1,5 kBytes) des Masterstartroutinencode zu generieren.

MBR-Hauptplatinen-ID-Wert 134: Das Datensegment beinhaltet einen Wert, wie eine Kette von Wörtern, die kompatible Hauptplatinen-IDs definieren. Jedes Wort besteht aus einer 16-Bit-Hauptplatinen-ID und die Kette endet mit einem Wortwert Null. Wenn die Hauptplatinen-ID eines Systems mit dem Hauptplatinen-ID-Wert in der Masterstartroutine übereinstimmt, wie eines der Wörter in der Kette, ist das IBL-Medienbild kompatibel mit der System-Hauptplatine. Wenn die ID der Systemhauptplatinen-ID mit keinem Wort in der Kette übereinstimmt, ist das IBL-Medienbild nicht kompatibel mit der System-Hauptplatine.

MBR-Modell- und Untermodellwerte 136: Das Datensegment umfaßt Werte wie z.B. eine Wortkette, die kompatible Prozessoren definiert. Jedes Wort ist zusammengesetzt aus einem Modell- und einem Untermodellwert, und die Kette endet mit einem Wortwert Null. Wenn eines der Modell- und Untermodellwörter des Systems (abgespeichert im ROM) mit einem Wort in der Kette

übereinstimmt, ist das IBL-Medienbild kompatibel mit dem Systemprozessor. Wenn die ROM-Modell- und ROM-Untermodellwörter mit keinem Wort in der Kette übereinstimmen, ist das IBL-Medienbild nicht mit dem Systemprozessor kompatibel.

MBR-Abbildungslänge 138: Die IBL-Abbildungslänge wird auf die Anzahl der Medienbildblöcke initialisiert. Mit anderen Worten, wenn das BIOS-Bild in vier Blöcke unterteilt ist, ist die Abbildungslänge vier, was vier Blockzeiger/Längenfelder anzeigt. Üblicherweise ist diese Länge auf Eins gesetzt, da das Medienbild ein zusammenhängender 128k Block ist.

MBR-Mediensektorgröße 138: Dieser Wortwert wird auf die Mediensektorgröße in Bytes per Sektor initialisiert.

Medienbildblockzeiger 138: Der Medienbildblockzeiger zeigt auf einen Systembildblock im Medium. Im Normalfall gibt es nur einen einzigen Zeiger, weil das Medienbild als ein zusammenhängender Block abgespeichert ist. Auf einer IBL-Diskette stehen die Zeiger im Spurkopfsektor-Format; auf der Platte stehen die Zeiger im Relativ-Blockadressen-Format.

Medienbildblocklänge 138: Die Medienbildblocklänge zeigt die Größe (in Sektoren) des Blocks, der am entsprechenden Bildblockzeiger liegt. Bei einem 128k zusammenhängenden Medienbild, einschließlich Platz für BASIC, ist dieses Feld auf 256 gesetzt, und zeigt damit an, daß der BIOS-Block 256 Sektoren besetzt (512 Bytes/Sektor), beginnend mir dem Ort, auf den der Medienbildblockzeiger zeigt.

Nehmen wir jetzt Bezug auf die Fig. 6A - D; hier wird ein detailliertes Flußdiagramm des Betriebs der IBL-Routine gezeigt. Unter normalen Umständen lädt die IBL-Routine die Masterstartroutine von der Systemfestplatte an einer bestimmten Adresse in den RAM, und vektorführt den Systemprozessor

dahin, mit dem Abarbeiten des Code-Segments der Masterstartroutine zu beginnen. Die IBL-Routine enthält auch Vorkehrungen für einen Diskettenvorgabemodus, in dem die Masterstartroutine von der Diskette geladen werden kann. Die IBL-Routine erlaubt jedoch nicht, daß der Diskettenvorgabemodus ausgeführt wird, wenn das System das IBL-Medium auf der Systemfestplatte enthält und ein gültiges Paßwort im NVRAM steht. Der Anwender hat die Option, das Paßwort in den NVRAM zu setzen. Der Zweck der Sperre der Ausführung des Diskettenvorgabemodus ist, zu verhindern, daß ein nichtzugelassenes BIOS-Bild von der Diskette geladen wird. Mit anderen Worten, der Diskettenvorgabemodus wird nur dann ausgeführt, wenn eine Systemfestplatte nicht betriebsbereit ist und der Anwender (durch Nichteingeben des Paßworts) seinen Wunsch gezeigt hat, von der Diskette laden zu können. Wenn die IBL-Routine die Masterstartroutine von keinem der beiden Medien laden kann, wird eine Fehleranzeige generiert und das System bleibt stehen.

Nehmen wir jetzt Bezug auf Fig. 6A; unter normalen Umständen enthält das System eine Systemfestplatte, die die IBL-Routine initialisiert, Schritt 150. Nehmen wir für Darstellungszwecke an, daß die Festplatte auf Laufwerk C des Personalcomputersystems konfiguriert ist. Nehmen wir ferner an, daß Laufwerk A als Diskettenlaufwerk bestimmt ist. Die IBL-Routine prüft dann Laufwerk C, um festzustellen, ob es IBL-Medien enthält, Schritt 152. Hier wird auf Fig. 6B hingewiesen, die diesen Vorgang detailliert beschreibt. Die IBL-Routine startet mit dem Lesen von der Festplatte an den letzten drei Sektoren und fährt mit dem Lesen, von 99 Sektoren fort oder bis eine gültige Masterstartroutine gefunden wird, wobei sie den Medienzeiger dekrementiert. Wenn eine Masterstartroutine gefunden wird, wird sie auf System-Hauptplatinen- und Prozessor-Kompatibilität überprüft, Schritt 156. Wenn sie nicht Hauptplatinen- oder Prozessor-kompatibel ist, wird ein Fehler gemeldet,

Schritt 158. Wieder zurück zu Schritt 152, wenn keine Masterstartroutine in den letzten 99 Sektoren der Festplatte gefunden wird (primäre Hardfile) wird ein Fehler gemeldet, Schritt 154.

Wieder zurück zu Schritt 156; wenn eine Masterstartroutine gefunden wird, werden eine Reihe von Gültigkeitsprüfungen durchgeführt, um festzulegen, ob die Masterstartroutine kompatibel mit dem Rechnersystem ist. Zusätzlich wird die Konfiguration des Systems geprüft. Hier wird auf Fig. 6D hingewiesen, die diesen Vorgang in weiteren Einzelheiten offenbart. Wenn die Startroutine mit Hauptplatinen-ID, Modell und Untermodell kompatibel ist, und ferner die Systemkonfiguration nicht verändert wurde, wird die Masterstartroutine geladen und das Codesegment der Masterstartroutine wird ausgeführt, Schritt 160.

Wieder zurück zu den Schritten 154 und 158; wenn beim Laden der Masterstartroutine von der Festplatte ein Fehler auftritt oder wenn keine Festplatte verfügbar ist, bestimmt die IBL-Routine, ob im NVRAM ein gültiges Paßwort steht, Schritt 162. Dieses Paßwort legt fest, ob das BIOS-Bild von der Diskette geladen werden kann. Hier wird darauf hingewiesen, daß das Paßwort nur dann existiert, wenn es vom Anwender durch Fahren eines festgelegten Dienstprogramms installiert wurde. Wenn im NVRAM ein Paßwort steht, kann das BIOS-Bild nicht von der Diskette geladen werden, Schritt 164. Dadurch kann der Anwender die Integrität der Operation des Systems sichern, indem er bewirkt, daß das System nur mit dem BIOS-Bild auf der Festplatte geladen werden kann. Das Paßwort kann in der Form einer Zeichenkette eingegeben werden, die im NVRAM gespeichert ist.

Wieder zu Schritt 162; wenn im NVRAM kein gültiges Paßwort steht, wodurch ermöglicht wird, das BIOS-Bild von der Dis-

kette zu laden, initialisiert die IBL-Routine das Disektten-Untersystem, Schritt 166. Die IBL-Routine bestimmt dann, ob das Laufwerk A das IBL-Medium auf einer Diskette enthält, Schritt 168. Wenn das Laufwerk A kein IBL-Medium enthält, wird eine Fehlermeldung generiert, um den Anwender zu unterrichten, daß eine ungültige Diskette in das Laufwerk eingelegt wurde, Schritt 170. Das System hält an, Schritt 172. Wieder weisen wir hier auf Fig. 6C hin, in der Schritt 168 eingehender diskutiert wird.

Wieder zurück zu Schritt 168; nachdem Laufwerk A auf das IBL-Medium geprüft wurde, wird die Masterstartroutine in den RAM geladen und das in der Masterstartroutine eingeschlossene Codesegment wird ausgeführt, Schritt 160. Es ist wichtig zu wissen, daß die IBL-Routine für die Diskette die Gültigkeitsüberprüfungen nicht durchführt, die für das Festplattensystem vorgesehen sind. Der Grund für das Fehlen der Gültigkeitsüberprüfungen ist, daß ein nichtkompatibles IBL-Bild von der Diskette geladen werden kann. Wenn z.B. ein neuer Prozessor zum System hinzugefügt wird, wird ein neues BIOS-Bild auf einer Diskette eingefügt. Da ein neuer Prozessor beim Laden von einer Festplatte Gültigkeitsfehler verursacht, sieht die IBL-Routine die Möglichkeit vor, durch Laden des BIOS-Bilds von einer Diskette diese Tests zu umgehen.

Fassen wir zusammen: Die Masterstartroutine wird auf Kompatibilität mit dem System geprüft durch Vergleichen auf Übereinstimmung der System-Hauptplatinen-ID und Prozessor-Modell-/Untermodell-Werte mit den Startroutinenwerten. Für die Platte wird diese Prüfung zunächst in der IBL-Routine 72 durchgeführt und dann wieder in der IBL-Startroutine. Die erste Prüfung (in der IBL-Routine) wird ausgeführt, um sicherzugehen, daß die Startroutine kompatibel mit dem System ist; die zweite Prüfung (in der Startroutine) wird ausgeführt, damit ein kompatibler ROM die Steuerung an die Start-

routine übergeben hat. Wir weisen darauf hin, daß die in der Platten-Startroutine durchgeführte Prüfung für einen kompatiblen ROM nie fehlschlagen kann, weil die IBL-Routine die Kompatibilität bereits überprüft hat. Im Gegensatz dazu wird die erste Kompatibilitätsprüfung für die Diskette nicht durchgeführt. Die Hauptplatinen/Prozessorkompatibilität wird nur während der Diskettenstartroutinen-Durchführung überprüft. Dieses Verfahren läßt künftige Modifikationen beim Laden eines neuen BIOS-Bilds von einer Referenzdiskette zu.

Im Hinblick auf die Beschreibung der IBL-Routine der Fig. 6A geht die Erklärung jetzt über auf ein umfassendes und volles Verständnis der oben diskutierten Gültigkeitstests. Nehmen wir Bezug auf Fig. 6B; dort wird ein detailliertes Flußdiagramm des Schritts 152 in Fig. 6A gezeigt, um festzustellen, ob eine gültige Masterstartroutine auf Laufwerk C vorhanden ist. Der Prozeß beginnt mit dem Einholen der Laufwerksparameter, um die IBL-Routine auf das Laufwerk C zugreifen zu lassen, Schritt 200. Eine IBL-Ladestelle wird auf die letzten drei Sektoren von der Platte gesetzt (die letzten drei Sektoren enthalten in der Regel die Masterstartroutine), Schritt 202. Eine Ladezählung, die die Anzahl der Versuche zum Lesen einer Masterstartroutine von der Platte anzeigt, wird auf 1 gesetzt, Schritt 204. Drei Sektoren werden an der IBL-Ladestelle von der Platte gelesen, Schritt 206. Etwaige Plattenlaufwerkfehler werden gefunden und wenn ein Plattenlaufwerk-Lesefehler vorkommt wird er gemeldet, Schritte 208-210. Dann kehrt der Prozeß mit einer Fehleranzeige zurück, Schritte 212-214.

Nehmen wir wieder Bezug auf Schritt 208; wenn kein Laufwerkfehler auftritt, wird der Plattensatz nach der Masterstartroutine-Signatur durchsucht, Schritt 216. Die Startroutine-Signatur, wie z.B. die Buchstaben "ABC", werden verglichen mit den drei ersten Bytes des Plattensatzes. Wenn der

Plattensatz eine gültige Startroutine-Signatur enthält (Buchstaben "ABC") und die vom Plattensatz in den Speicher geladene Prüfsumme gleich ist der Prüfsumme der Startroutine, wird der Plattensatz als gültige Startroutine ohne Fehler angezeigt, Schritt 218. Der Prozeß springt dann wieder zurück, Schritt 214.

Nehmen wir wieder Bezug auf Schritt 216; wenn die Startroutine-Signatur der Prüfsumme ungültig ist, wird die Ladezählung um 1 inkrementiert, Schritt 220. Die Ladezählung wird dann mit einer vorbestimmten Konstanten wie z.B. 99 verglichen, Schritt 222. Wenn 99 Versuche des Lesens einer Startroutine fehlgeschlagen sind, wird ein Fehler angezeigt, und der Prozeß springt zurück, Schritte 224, 212 und 214. Wenn weniger als 99 Versuche, eine Startroutine zu lesen, vorgekommen sind, wird die IBL-Ladestelle um Eins dekrementiert und drei neue Sektoren werden von der neuen Ladestelle aus gelesen, Schritte 226 und 206. Wenn also keine gültige IBL-Startroutine aus den letzen 99 Sektoren (das ist gleichbedeutend mit 33 Kopien) geladen werden kann, dann wird eine Fehlerbedingung gesetzt und die Steuerung kehrt zur IBL-Routine zurück.

Gehen wir jetzt zu Fig. 6C; hier wird ein detailliertes Flußdiagramm zum Laden der Masterstartroutine von der Diskette in Laufwerk A gezeigt. Zunächst werden die Diskettenlaufwerk-Parameter für den Zugriff auf Laufwerk A abgerufen, Schritt 230. Die IBL-Ladestelle wird auf die letzten 3 Sektoren der Diskette gesetzt (Zylinder-, Kopf- und Sektor-Format), Schritt 232. Die letzten 3 Sektoren werden gelesen, Schritt 234. Wenn ein Diskettenlaufwerkfehler erfaßt wird, wird ein Fehler gemeldet, Schritte 236-238. Ein Fehlerzustand wird gesetzt und die Steuerung geht wieder auf die IBL-Routine über, Schritte 240-242.

Wieder zu Schritt 236; wenn kein Laufwerkfehler erfaßt wird, wird die Diskettenaufzeichnung auf die Startroutine-Signatur geprüft und die Prüfsumme wird berechnet, Schritt 244. Wenn die Startroutine-Signatur fehlt oder die Prüfsumme ungültig ist, wird ein Fehler gemeldet und die Steuerung geht wieder auf die IBL-Routine über, Schritte 244, 246, 240 und 242. Wenn eine gültige Steuerroutine-Signatur und eine gültige Prüfsumme gefunden werden, wird eine Anzeige gesetzt und die Steuerung geht wieder auf die IBL-Routine über, Schritte 248 und 242. Es wird darauf hingewiesen, daß beim Laden von einer Diskette die IBL-Routine das Medium nicht durchsucht wie beim Laden von der Festplatte. Deshalb muß das IBL-Medium an einer bestimmten Stelle der Diskette abgespeichert sein.

Schließlich zeigt Fig. 6D, wie die IBL-Routinen die System-Hauptplatinen- und Prozessor-Kompatibilität und die geeignete Systemkonfiguration testen. Die Masterstartroutine wird auf Kompatibilität mit der System-Hauptplatine geprüft durch Vergleichen des Startroutinen-Hauptplatinen-ID-Werts mit der System-Hauptplatinen-ID, die vom Systemprozessor gelesen wird, Schritt 260. Wenn die System-Hauptplatinen-ID nicht mit dem Systemladeroutine-ID-Wert übereinstimmt, zeigt das an, daß diese Masterstartroutine mit dieser Hauptplatine nicht kompatibel ist. Ein Fehler wird gemeldet und die Steuerung kehrt zur IBL-Routine zurück, Schritte 262, 264 und 266.

Wenn die Masterstartroutine kompatibel mit der Hauptplatine ist, wird die Masterstartroutine auf Kompatibilität mit dem Prozessor geprüft, Schritt 268. Der Startroutinen-Modellwert und -Untermodellwert werden verglichen mit dem Modellwert bzw. Untermodellwert, die im ROM gespeichert sind. Wenn die Werte nicht übereinstimmen, zeigt das an, daß wahrscheinlich ein neuer Prozessor eingesetzt wurde und diese Startroutine nicht mit dem neuen Prozessor kompatibel ist. Ein Fehler wird gemeldet und die Steuerung geht zurück zur IBL-Routine,

Schritte 270, 264 und 266. Wenn die Masterstartroutine mit der Hauptplatine und dem Prozessor kompatibel ist, prüft der Prozeß, ob der NVRAM zuverlässig ist, Schritt 272. Wenn der NVRAM unzuverlässig ist, wird ein Fehler gemeldet und die Steuerung geht zurück zur IBL-Routine, Schritte 274 und 266. Wenn der NVRAM zuverlässig ist, wird die Systemkonfiguration geprüft, Schritt 276. Eine Veränderung der Systemkonfiguration wird angezeigt, wenn die in NVRAM abgespeicherten Modell- und Untermodellwerte nicht mit den im ROM abgespeicherten Modell- und Untermodellwerten übereinstimmen. Hier wird darauf hingewiesen, daß dieser letzte Vergleich nur einen Konfigurationsfehler anzeigt. Wenn ein Konfigurationsfehler gemeldet wird, wird eine Fehlermeldung an den Anwender generiert. Dieser Fehler unterrichtet den Anwender, daß die Konfiguration des Systems sich seit dem letzen Mal, als ein 'SET Configuration' gefahren wurde, verändert hat. Der Anwender wird über die veränderte Konfiguration in Kenntnis gesetzt und die Steuerung geht zurück zur IBL-Routine, Schritte 278, 264 und 266. Dieser Fehler ist als solcher kein fataler Fehler, sondern unterrichtet den Anwender, daß SET Configuration (Konfigurationsprogramm) gefahren werden muß. Wieder unter Bezugnahme auf Schritt 276, wenn die System-Modell/Untermodellwerte übereinstimmen, wird eine Anzeige der Vergleichbarkeit gesetzt und die Routine springt wieder zurück, Schritte 276, 274 und 266. Auf diese Weise wird die Kompatibilität zwischen der Masterstartroutine und dem System geprüft, zusammen mit der Feststellung, ob die Systemkonfiguration verändert wurde.

Nachdem die IBL-Routine die Masterstartroutine in den RAM geladen hat, überträgt sie die Steuerung an die MBR-Code-Startadresse. Nehmen wir Bezug auf Fig. 7; das ausführbare Codesegment der Masterstartroutine überprüft zunächst das Startroutinemuster mit dem ROM Muster, Schritt 300. Wenn das Muster in der Masterstartroutine nicht mit dem Muster im ROM

übereinstimmt, wird ein Fehler generiert und das System bleibt stehen, Schritte 302 und 305. Die Überprüfung auf Gleichheit zwischen ROM und Startroutinenmuster stellt sicher, daß die entweder von der Platte oder von der Diskette geladenen Masterstartroutine kompatibel mit dem ROM und der Hauptplatine ist. Nehmen wir wieder Bezug auf Schritt 300; wenn das Muster im ROM mit dem Muster in der Startroutine übereinstimmt, vergleicht der MBR-Code den System-Hauptplatinen-ID-Wert, den Modell- und Untermodellwert mit den entsprechenden Masterstartroutinewerten, Schritt 304. Dieser Prozeß wurde in weiteren Einzelheiten anhand Fig. 6D diskutiert. Wenn die Werte nicht übereinstimmen, ist die Masterstartroutine nicht kompatibel mit der System-Hauptplatine im Prozessor, oder die Systemkonfiguration wurde verändert, Schritt 306. Das System bleibt stehen, wenn die IBL-Routine nicht kompatibel mit der Hauptplatine, den Modell- oder Untermodellwerten ist, Schritt 305.

Nehmen wir wieder Bezug auf Schritt 304; Wenn der System-Hauptplatinen-ID-Wert, die Modell- und Untermodellwerte mit den entsprechenden Masterstartroutinewerten übereinstimmen, lädt der MBR-Code das BIOS-Bild vom angewählten Medium in den System-RAM, Schritt 308. Wen ein Mediumladefehler beim Lesen der Daten auftritt, Schritt 310, wird ein Fehler generiert und das System bleibt stehen, Schritt 312 und 305. Nehmen wir Bezug auf Schritt 310; wenn kein Medium-Ladefehler auftritt, wird eine Prüfsumme berechnet aus dem BIOS-Bild im Speicher, Schritt 314. Wenn die Prüfsumme ungültig ist, wird ein Fehler generiert und das System bleibt stehen, Schritt 318 und 305. Nehmen wir Bezug auf Schritt 316, wenn die Prüfsumme gültig ist, werden die Systempartitionszeiger abgespeichert, Schritt 320, und der Systemprozessor wird auf POST Stufe II vektorgeführt, um mit dem Laden des Systems anzufangen, Schritt 322.

Nehmen wir Bezug auf Fig. 8; dort wird ein Blockdiagramm eines intelligenten Platten-Controllers 350 zum Steuern der Datenbewegung zwischen dem Plattenlaufwerk 351 und dem Systemprozessor gezeigt. Hier muß darauf hingewiesen werden, daß der Platten-Controller 350 in die Adapterkarte 60 integriert sein kann während das Plattenlaufwerk 251 im Laufwerk 62 in Fig. 2 enthalten sein kann. Ein geeigneter Platten-Controller 350 ist ein SCSI-Adapter mit der Teilenummer 33F8740, hergestellt von der International Business Machines Corporation. Hier wird verstanden, daß der Platten-Controller 350 einen Mikroprozessor 352 beinhaltet, der unter seinem eigenen internen Zeitgeber arbeitet zur Steuerung seiner internen Operationen sowie auch über Schnittstelle mit anderen Elementen des Platten-Untersystems und mit dem Systemprozessor in Verbindung steht. Der Mikroprozessor 352 ist über einen Befehlsbus 354 mit einem Nurlesespeicher (ROM) 356 gekoppelt, der die Befehle speichert, die der Platten-Controller 350 abarbeitet, um die Datenbewegungen zwischen dem Plattenlaufwerk und dem Systemprozessor zu verarbeiten und zu steuern. Ebenso muß verstanden werden, daß der Platten-Controller 350 Speicher mit wahlfreiem Zugriff umfassen kann, die an den Mikroprozessor 352 zum Speichern und Abrufen von Daten angekoppelt sind. Die Datenbwegungen zwischen dem Platten-Controller 350 und dem Systemprozessor wird bewirkt über den Datenbus 358 und den Befehlsbus 360. Ein Rückstellsignal auf Leitung 362 stellt die Platten-Controller-Logik zurück oder initialisiert sie bei der Einschaltsequenz oder während einer Systemrückstellung. Das Rückstellsignal wird generiert durch die Hauptplatinen-Logik und kann die Form eines Kanalrückstellsignals annehmen, wie in der IBM Mikrokanal-Architektur vorgesehen ist und wie in "IBM PERSONAL SYSTEM/2 Seminar Proceedings", Bd. 5 Nr. 3, Mai 1987, beschrieben ist, wie von der International Business Machines Corporation Entry Systems Division veröffentlicht wurde. Ferner kann das Rückstellsignal durch das BIOS wirksam initiiert werden durch Ausgabe einer beson-

deren Bit-Konfiguration an einen E/A-Port des Systemprozessors, an den die Hauptplatinenlogik angeschlossen ist.

Wie wohlbekannt ist, sieht der Mikroprozessor 352 alle Schnittstellenverbindungen und Zeitgebersignale vor, um eine wirksame Datenübertragung zwischen dem Plattenlaufwerk und dem Systemprozessor zu erzeugen. Zwecks Klarheit werden nur solche Signale dargestellt, die für das Verständnis der Erfindung wichtig sind. Verstanden werden muß ferner, daß auch andere Signale und Leitungen, wie der Datenbus 364, benutzt werden, aber hier nicht dargestellt sind, weil sie für das Verständnis der vorliegenden Erfindung nicht von Bedeutung sind. Ferner muß darauf hingewiesen werden, daß nur solche im ROM 356 gespeicherte Programme und Routinen, die für das Verständnis der vorliegenden Erfindung wichtig sind, im Hinblick auf Fig. 9 erklärt werden.

Nehmen wir jetzt Bezug auf Fig. 9; dort ist ein Flußdiagramm gezeigt, das die durch die Operation von im ROM 356 gespeicherte Routinen bewirkte Lese-, Schreib- und Schutz-Funktionen des Platten-Controllers als Diagramm darstellt. Im Betrieb wird eine Plattenanweisung durch den Systemprozessor initialisiert und zum Platten-Controller 350 übertragen. Der Platten-Controller empfängt und interpretiert die zur Ausführung der bezeichneten Operation auszuführenden Anweisungen, Schritt 400. Der Platten-Controller bestimmt zunächst, ob es sich um eine Schreiboperation handelt, in der Daten von Systemprozessor auf der Plattenlaufwerk-Hardware gespeichert werden, Schritt 402. Wenn die Anweisung eine Schreibanweisung ist, werden die Daten aus dem Systemprozessor im relativen Blockadressen-Format (RBA) empfangen.

Vor Fortsetzung der obigen Diskussion sollte eine kurze Erklärung des relativen Blockadressenformats, das für eine Massenspeichervorrichtung wie eine Platte angewandt wird,

vorteilhaft wiederholt werden. RBA ist ein Schema, in dem Massenspeicherungsdaten in Blöcken vorgegebener Größe durch aufeinanderfolgende Nummern, d.h. in individuell definierbaren aufeinanderfolgenden Datenblöcken adressiert werden. Wenn wir z.B. eine Blockgröße von 1024 Bytes annehmen, kann der Systemprozessor auf einer 10 Megabyte-Platte etwa 10 000 Blöcke adressieren. Das heißt, der Systemprozessor kann das Plattenmedium als N Blöcke adressieren, wobei N von 0 bis 9 999 läuft. Es wurde entdeckt, daß die Anwendung von RBA eine sehr schnelle und wirksame Methode zum Adressieren von Massenspeicherungen in dem Betriebssystemtyp vorsieht, der für Personalcomputersysteme der vorliegenden Erfindung benutzt wird.

Zwecks Bequemlichkeit sollen die folgenden Annahmen eingeführt werden: Erstens, die Platte kann insgesamt N Blöcke aufnehmen; zweitens, der Systemprozessor überträgt einen K-Block, wobei K größer oder gleich 0, und kleiner oder gleich (N-1) ist; drittens, der Platten-Controller kann einen maximal adressierbaren Block M setzen, der Zugriff auf Datenblöcke erlaubt, wobei K kleiner M ist, und den Zugriff auf Datenblöcke verweigert, wenn K größer oder gleich M ist. Hier sei darauf hingewiesen, daß durch Setzen von M kleiner als N ein schutzfähiger Bereich auf der Platte von M bis N-1 Blöcken generiert wird. Dieses Merkmal läßt zu, daß das IBL-Medium geschützt wird, wie nachstehend noch erklärt wird.

Unsere Diskussion soll nun unter Bezugnahme auf Fig. 9 fortgesetzt werden; die Daten werden von der Platte im RBA-Format aufgenommen, Schritt 404. Dann bestimmt der Platten-Controller, ob der aufgenommene Block K kleiner als der Maximalblockwert M ist, wobei M kleiner als N ist, Schritt 406. Wenn K kleiner als M ist, dann wandelt der Controller das RBA-Format um in das besondere Format für die Massenspeichervorrichtung, wie das Zylinder-Kopf-Sektor (Cylinder-Head-

Sector - CHS)-Format für eine Festplatte um, Schritt 408. Zum Beispiel könnte der Platten-Controller durch Verwenden einer Nachschlagetabelle RBA-Adressen in unverwechselbare Zylinder-Kopf-Sektor-Stellen umwandeln. Ein weiteres Verfahren wäre die Verwendung einer Umwandlungsformel zum Umwandeln von RBA in CHS. Zum Beispiel, für eine Platte mit einem Kopf, 64 Zylinder und 96 Sektoren: Kopf = 0, Zylinder = Quotient RBA/96, und Sektoren = Rest von RBA/96. Nach Umwandlung des RBA-Formats in ein CHS-Format werden die Daten an der umgewandelten CHS-Stelle auf die Platte geschrieben, Schritt 410. Dann wartet der Platten-Controller auf eine weitere Anweisung vom Systemprozessor, Schritt 412.

Wieder zurück zu Schritt 406; wenn die eingegangene RBA größer ist als der gesetzte Maximal-RBA-Wert, wird der Zugriff verweigert, Schritt 414. Das heißt, wenn K größer oder gleich M ist, wird der K-Block nicht auf die Platte geschrieben. Hier ist anzumerken, wenn das IBL-Medium in den Blöcken von M bis N-1 gespeichert ist, wird das IBL-Medium gegen Schreiben geschützt.

Wider zurück zu Schritt 402; wenn die Anweisung vom Systemprozessor keine Schreibanweisung ist, dann wird geprüft, ob es sich um eine Leseanweisung handelt, Schritt 416. Wenn die Anweisung eine Leseanweisung ist, sendet der Systemprozessor das RBA-Format für die angeforderten Daten, Schritt 418. Dann bestimmt der Platten-Controller, ob die gewünschte RBA (K) kleiner ist als die maximale gesetzte RBA (M). Wenn die gewünschte RBA (K) kleiner ist als die gesetzte maximale RBA (M), wandelt der Platten-Controller die RBA in das geeignete CHS-Format um und liest die Daten von der Platte, Schritte 422 und 424. Die Daten werden dann auf den Systemprozessor übertragen, Schritt 412.

Nun wieder zurück zu Schritt 420; wenn die eingegangene RBA (K) größer oder gleich der gesetzten maximalen RBA (M) ist, wird der Zugriff verweigert, Schritt 426. Wenn das IBL-Medium zwischen M Blöcke und (N-1) Blöcke gespeichert ist, wird der Zugriff auf diesen Bereich verweigert. Hier ist anzumerken, daß unter diesen Umständen das IBL-Medium auch gegen Kopieren geschützt ist.

Wieder zurück zu Schritt 416; wenn die Anweisung keine Schreib- oder Leseanweisung ist, wird sie auf eine Maximum-RBA-Setzen-Anweisung geprüft, Schritt 428. Diese Anweisung erlaubt, daß der Platten-Controller einen schutzfähigen Bereich oder eine Partition auf der Plattenlaufwerk-Hardware erzeugt. Diese Anweisung ermöglicht, daß der Platten-Controller M zwischen 0 und N Blöcke setzt, Schritt 430. Hier ist es wichtig zu bemerken, daß beim Rückstellen des Platten-Controllers (über das Rückstellsignal) M so gesetzt wird, daß die Maximalzahl Blöcke verfügbar ist. Das heißt, wenn der Platten-Controller rückgestellt wird, ist M=N. Im wesentlichen wird der Schutz für den schutzfähigen Bereich durch Rückstellen des Platten-Controllers aufgehoben und jetzt ist der Zugriff auf diesen Bereich möglich. Jedoch wird, sobald die Anweisung Maximum-RBA-Setzen ausgeführt ist, nur ein Rückstellen oder eine andere Maxiumum-RBA-Setzen-Anweisung einen Zugriff auf den schutzfähigen Bereich möglich machen. Man kann sich das Setzen der Maximum-RBA begrifflich als Aufstellen eines Zauns vorstellen, der den Zugriff auf den Bereich oberhalb des Zauns verhindert während er den Zugriff auf den Bereich unterhalb des Zauns zuläßt. Dann kehrt der Platten-Controller wieder zurück und wartet auf eine weitere Anweisung, Schritt 412.

Nehmen wir wieder Bezug auf Schritt 428; wenn die Anweisung keine Anweisung zum Lesen, Schreiben oder Setzen von RBA auf Maximum ist, wird sie auf eine andere Platten-Controller-An-

weisung geprüft und ausgeführt, Schritt 432. Diese Anweisungen benutzen den gesetzten RBA-Maximal-Wert, sind jedoch nicht bedeutsam für das Verständnis der vorliegenden Erfindung und werden hier zwecks Kürze nicht dargestellt. Dann kehrt der Platten-Controller für eine weitere Anweisung auf Wartestellung zurück, Schritt 412.

Im Hinblick auf die weitere Diskussion geht die Erklärung jetzt auf die Operation Laden und Schützen des IBL-Mediums über. Im allgemeinen wird der Platten-Controller mit dem IBL-Medium von einem Kaltstart (power-on - Einschalten) oder einem Warmstart (Ctrl-Alt-Del - Strg-Alt-Entf - [Affengriff]) rückgestellt. Das hat die Wirkung, daß die Maximum-RBA (M) auf N gesetzt wird, d.h. der Zaun wird weggeräumt und Zugriff auf das IBL-Medium ist möglich. Das ist erforderlich damit das System das IBL-Medium laden kann, um die Operation zu beginnen. Sobald das IBL-Medium geladen und abgearbeitet ist, wird der Zaun wieder aufgebaut (Maximum-RBA wird unter das IBL-Medium gesetzt), damit der Zugriff auf das auf der Platte gespeicherte IBL-Medium verhindert wird.

Nehmen wir jetzt Bezug auf Fig. 10; hier wird ein Block-Flußdiagramm gezeigt, das den Schutz des IBL-Mediums bewirkt. Von einem Power-on-Zustand wird das System initialisiert und das BIOS beginnt seine Tätigkeit in der Hauptplatinenkartenlogik, um einen Rückstellzustand an den Platten-Controller zu schicken, Schritte 450 und 452. Das Rückstellsignal setzt den Zaun tief und ermöglicht es dem Systemprozessor, auf das vorher auf der Platte gespeicherte IBL-Medium im Bereich vom M Blöcken bis N Blöcken Zugriff zu nehmen. Das System lädt das IBL-Medium, wie bereits unter Bezugnahme auf Fig. 4-7 beschrieben, Schritt 454. Während der IBL-Ladesequenz wird die POST-Stufe II ausgeführt, Schritt 456. Eine der Aufgaben der POST-Stufe II ist die Durchführung der Anweisung Maximum RBA Setzen, wobei die Maximum-RBA auf den ersten Block des IBL-

Mediums, bezeichnet als M, gesetzt wird, Schritt 458. M hängt ab vom Partitionstyp (keine, teilweise oder voll) wie oben bereits erklärt. Das setzt nämlich den Zaun, der den Zugriff auf das IBL-Medium verhindert, während er den Zugriff auf andere Plattenbereiche zuläßt. Dann wird das Betriebssystem ganz normal gebootet, Schritt 460.

Wenn das System mit einem Warmstart, Ctrl-Alt-Del - Strg-Alt-Entf, angelassen wird, bekommt die Hauptplatinenlogik den Befehl, den Platten-Controller durch die POST-Stufe II rückzustellen, Schritte 462 und 464. Das bewirkt Rücknahme des Zauns. Unter diesen Umständen wird das IBL-Medium nicht wieder geladen, da es ja im RAM bereits vorhanden ist. Da aber der Schutz für das IBL-Medium ausgeschaltet ist, muß die POST-Stufe II erneut abgearbeitet werden, um den Zaun wieder zu setzen, Schritte 456 und 568. Der Zaun wird gesetzt und schützt das IBL-Medium, und das System wird ganz normal neu gebootet, Schritt 460.

Das IBL-Medium wird geschützt durch Adressieren der Massenspeicherung in Blöcken, und das Setzen eines Maximum-Blocks, auf die das System bei normaler Operation zugreifen kann. Das IBL-Medium wird der Reihe nach in diesen Blöcken zwischen dem maximal zugreifbaren Block und der Gesamtzahl der Blöcke gespeichert, die vom Plattenlaufwerk unterstützt werden. Ein Rückstellsignal, das an den Platten-Controller geschickt wird, eliminiert den zugreifbaren Maximal-Block, um dem System zu ermöglichen, das IBL-Medium zu adressieren. Das Rückstellsignal wird während des Einschalten oder des Warmstarts generiert, damit auf das IBL-Medium zum Booten des Systems zugegriffen werden kann.

Nehmen wir jetzt Bezug auf Fig. 11; das Flußdiagramm beschreibt den Prozeß, den die POST-Stufe II verfolgt, um das System-Referenzdiskettenbild von der Systempartition auf der

Festplatte 62 zu laden. Vor dem Urladen eines Betriebssystems, wie DOS oder OS/2 von IBM, stellt POST sicher, welcher Systempartitionstyp auf dem IBL-Medium vorhanden ist, Schritt 500. Dann fragt POST die Festplatte 62 ab nach dem Wert der letzten Blockadresse, Schritt 502. Dann berichtigt POST den als letzte Blockadresse erhaltenen Wert, um die Größe der Systempartition zu berücksichtigen, Schritt 504. Das geschieht durch Subtrahieren der Anzahl der Blöcke in der Systempartition von der physikalisch letzten Blockadresse auf der Festplatte 62. POST speichert den berichtigten Wert als logisch letzte Blockadresse, Schritt 506. Dadurch hat POST dem BIOS einen Mechanismus gegeben, um von der Systempartition anstatt vom Anfang der Festplattenpartition aus zu booten. Näheres hierzu siehe Fig. 13.

Gehen wir weiter in Fig. 11; POST Stufe II prüft die augenblicklichen Inhalte des POST-Pfad-Flag, Schritt 508. Das POST-Pfad-Flag ist ein Mechanismus, der von POST benutzt wird, um den Typ des Pfads durch POST verfolgen zu können, zum Beispiel, einen Anfangs-Einschaltpfad (power-on) im Vergleich zu einem Warmstart-Neuladen. Ein Warm-Neustart wird in der Regel durch eine Tastenkombination Ctrl-Alt-Del (Strg-Alt-Entf) bewirkt. Wenn der augenblickliche Wert des POST-Pfad-Flag Überschreiben der Systempartitions-Boot-Verfahren anzeigt, setzt POST-Stufe II das Systempartitions-Boot-Flag auf Falsch und zeigt damit an, die Systempartition nicht zu booten, Schritt 510. POST-Stufe II schützt dann die Systempartition durch Anweisung an das BIOS, das Schutzmittel auf der Urlade-Festplatte auf der Gundlage des in Schritt 506 berechneten Werts zu aktivieren, Schritt 511. Das heißt, der Zaun wird auf den Adressenzeiger gesetzt, der in Schritt 506 berechnet wurde. Somit ist die Systempartition gegen eine unbeabsichtigte Zerstörung geschützt. Anschließend ruft POST-Stufe II den Urlader INT 19h auf, das Urladen des Betriebssystems anlaufen zu lassen, Schritt 512.

Nehmen wir wieder Bezug auf Schritt 508; wenn das POST-Pfad-Flag kein Überschreiben der Systempartitions-Boot-Sequenz angibt, wird das POST-Pfad-Flag auf einen Warm-Boot-Pfad geprüft, wobei POST angibt, daß eine Ctrl-Alt-Del (Strg-Alt-Entf) Tastenkombination eingegeben wurde, Schritt 520. Wenn das Pfad-Flag keinen Warmstart-Boot anzeigt, bestimmt POST-Stufe II, ob beim Ausführen eines Kaltstarts Fehler aufgetreten sind, Schritt 522. Wenn keine Fehler aufgetreten sind, setzt POST-Stufe II ein Flag, das angibt, die Systempartition nicht zu booten, Schritt 510. Jetzt schützt POST-Stufe II die Systempartition durch Anweisung an das BIOS, das Schutzmittel zu aktivieren wie in Schritt 511 angegeben ist, gefolgt vom Aufrufen des Urladers, Schritt 512.

Wieder zurück zu Schritt 522; wenn POST-Stufe II Fehler bei der Abarbeitung entdeckt, setzt es das System-Partitions-Boot-Flag auf Wahr, Schritt 526. POST-Stufe II schützt dann die Systempartition durch Anweisung an das BIOS, das Schutzmittel zu aktivieren wie in Schritt 511 gezeigt wird. Anschließend ruft POST-Stufe II den Urlader 512 auf, den Betriebssystem-Urladevorgang einzuleiten.

Wieder zurück zu Schritt 520; wenn die Ctrl-Alt-Del (Strg-Alt-Entf) Tastenkombination eingegeben wurde, überprüft POST-Stufe II, ob der Anwender die Tastenkombination Ctrl-Alt-Ins (Strg-Alt-Einfg) eingegeben hat. Die Tastenanweisung Ctrl-Alt-Ins (Strg-Alt-Einfg) ruft das Urladen des Systemreferenzdisketten-Bilds 524 auf. Diese Sequenz erlaubt es dem Anwender, eine Urladeprozedur von der Systempartition zu erzwingen. Wenn nicht, setzt POST-Stufe II das Systempartitions-Boot-Flag auf Falsch und schützt die Systempartition durch Anweisung an das BIOS, das Schutzmittel zu aktivieren, wie in Schritt 511 angegeben ist. Dann ruft POST-Stufe II den Urla-

der INT 19h auf, den Betriebssystem-Urladevorgang einzuleiten, Schritt 512.

Gehen wir wieder zurück zu Schritt 524; wenn POST-Stufe II feststellt, daß der Anwender die Tastenkombination Ctrl-Alt-Ins (Strg-Alt-Einfg) eingegeben hat, setzt sie das Systempartitions-Boot-Flag auf Wahr und gibt dadurch an, die Systempartition zu booten, Schritt 526. Dann schützt POST-Stufe II die Systempartition durch Anweisung an das BIOS, das Schutzmittel zu aktivieren wie in Schritt 511 angegeben ist, gefolgt vom Aufrufen des Urladers, Schritt 512.

An diesem Punkt hat POST-Stufe II festgelegt, ob entweder eine normale Bootsequenz oder ein Booten vom Systemreferenz-Diskettenbild in die Systempartition stattfinden soll. POST hat auch den Anfang der Systempartition festgelegt, so als ob diese eine logisch urladbare Partition sei, und hat das Schutzmittel aktiviert, um den Zugriff auf die Systempartition durch ein Programm zu verhindern, auf das man sich nicht verlassen sollte. Eine logisch urladbare Partition erscheint dem POST als erste Partition auf der Platte und ist daher urladbar. Jetzt ruft POST-Stufe II den Urlader auf.

Der Urlader wird benutzt, um die geeignete Urladevorrichtung anzuwählen und die Startroutine von der aktiven Partition einzulesen. Die Priorität der Boot-Laufwerke ist: Das erste Diskettenlaufwerk, gefolgt von der ersten Festplatte, wie z.B. die Bootfestplatte. Jedoch kann die Priorität der Vorgabe-Bootvorrichtungs-Reihenfolge durch Anwendung eines Dienstprogramms auf der Systemreferenzdiskette oder im Systemreferenzdiskettenbild in der Systempartition verändert werden. Der Urlader-Controller gibt dann die Steuerung an den ausführbaren Code in der Startroutine. Dieser bootet dann das gewünschte Betriebssystem oder Steuerprogramm.

Setzen wir die Diskussion im Hinblick auf Fig. 12 fort; hier wird ein Flußdiagramm gezeigt, das den logischen Fluß innerhalb des Urladers, INT 19h, zeigt. Zunächst prüft der Urlader, ob im ersten Diskettenlaufwerk eine Systemreferenzdiskette liegt, Schritt 600. Das Vorhandensein einer Systemreferenzdiskette im ersten Diskettenlaufwerk überschreibt alle weiteren Referenzdisketten. Mit anderen Worten, das Aufrufen der Systemreferenzdiskette überschreibt das Systemreferenzdiskettenbild in der System-Partition oder eine direkte Anforderung durch den Anwender zum Urladen des Betriebsystems, wenn POST-Fehler gefunden wurden. Als nächstes wird das Systempartitions-Boot-Flag geprüft, Schritt 620. Da die Systemreferenzdiskette eingelegt ist, steht das Systempartitions-Boot-Flag auf Falsch.

Da das Systempartitions-Boot-Flag auf Falsch steht, bestimmt der Urlader, ob ein Referenzdisketten-Booten erforderlich ist, Schritt 630. Da eine Systemreferenzdiskette im ersten Laufwerk liegt, weist der Urlader das BIOS an, zunächst das Schutzmittel für die Systempartition zu deaktivieren, Schritt 640. Dann errichtet der Urlader die Systempartition als Ursprung der Urladefestplatte durch Anwenden des Werts, der im Schritt 506 als logische Anfangsblockadresse berechnet wurde, Schritt 650. Jetzt ist die Systempartition ungeschützt. Der Urlader holt dann die Startroutine von der Systemreferenzdiskette und gibt die Steuerung an diese, Schritt 660. Die Startroutine bootet dann die Systemreferenzdiskette. Zum Beispiel kann ein Anwender einen neuen E/A-Adapter für ein neues Leistungsmerkmal ins System einfügen und will nun seine Adapterbeschreibungsdatei in die Systempartition installieren.

Wieder zurück zu Schritt 600; wenn im ersten Diskettenlaufwerk keine Referenzdiskette liegt, prüft der Urlader das Systempartitions-Boot-Flag, Schritt 612. Wenn das Flag einen

Betriebssystem-Urladevorgang anzeigt, überträgt der Urlader die Steuerung auf die wählbare Urladeroutine, Schritt 614. Die wählbare Urladeroutine entscheidet dann, von welcher physikalischen Vorrichtung sie booten will, und geht zu Schritt 620 über.

Dann wird auf das Systempartitions-Boot-Flag zugegriffen, um festzustellen, ob es gesetzt ist, Schritt 620. Wenn kein Systempartitions-Boot angefordert wird, legt der Urlader fest, ob ein Systemreferenzdisketten-Boot erforderlich ist, Schritt 630. Zum Beispiel kann in einem urladbaren Diskettenlaufwerk, das nicht das erste physikalische Diskettenlaufwerk ist, eine Systemreferenzdiskette liegen. Wenn keine Systemreferenzdiskette vorhanden ist, holt der Urlader die Betriebssystem-Startroutine und übergibt ihr die Steuerung, Schritt 660. Die Systempartition bleibt geschützt und das BIOS greift auf eine andere Partition zu, nämlich auf die Betriebssystempartition auf der Urlade-Festplatte.

Gehen wir wieder über auf Schritt 630; wenn ein Systemreferenzdisketten-Boot angefordert ist, weist der Urlader das BIOS an, das Schutzmittel für die Systempartition zu deaktivieren und die Systempartition als Ursprung der Urladefestplatte einzurichten durch Anwenden des in Schritt 506 als logische Startblockadresse errechneten Werts, Schritt 650. Der Urlader holt die Startroutine von der Referenzdiskette (in diesem Fall ist eine Referenzdiskette vorhanden) und bootet die Systemreferenzdiskette, Schritt 660. Die Systempartition ist ungeschützt und ist jetzt die aktive Partition auf der Festplatte. Das geschieht, um den Zugriff durch die Referenzdiskette zu ermöglichen. Wie früher erklärt, kann ein Anwender einen neuen E/A-Adapter in das System aufnehmen wollen und will jetzt seine Adapterbeschreibungsdatei in die Systempartition installieren.

Nehmen wir wieder Bezug auf Schritt 620; wenn das Systempartitions-Boot-Flag auf Wahr steht, weist der Urlader das BIOS an, das Schutzmittel für die Systempartition Schritt 640 zu deaktivieren und die Systempartition als Ursprung der Urladefestplatte einzurichten durch Anwenden des in Schritt 506 als logische Startblockadresse errechneten Werts, Schritt 650. Der Urlader holt dann die Startroutine vom Referenzdisketten-bild in der Systempartition und bootet das Systemreferenz-diskettenbild, Schritt 660. Die Systempartition ist ungeschützt und ist jetzt die aktive Partition auf der Boot-Festplatte.

Gehen wir zurück zu Schritt 612; wenn das Systempartitions-Boot-Flag ein Systempartitions-Booten anzeigt, prüft der Urlader auf eine gültige Startroutine auf der Systempartition, Schritt 616. Dieser Schritt beinhaltet das Überprüfen, daß die Systempartition eine volle Systempartition ist, die Startroutine-Signatur gültig ist, und eine Systemreferenz-diskettensignatur vorhanden ist. Wenn gültig, fragt der Urlader das Systempartitions-Boot-Flag ab, Schritt 620. Da dieses Wahr ist, weist der Urlader das BIOS an, das Schutzmittel für die Systempartition zu deaktivieren und die Systempartition als Ursprung der Bootfestplatte einzurichten durch Anwenden des in Schritt 506 als logische Startblockadresse errechneten Werts, Schritte 640 und 650. Der Urlader holt dann den Startroutine von der Systempartition und urladet das Systemreferenzdiskettenbild, Schritt 660. Die Systempartition ist ungeschützt und ist jetzt die aktive Partition auf der Boot-Festplatte.

Nehmen wir wieder Bezug auf Schritt 616; wenn keine gültige Startroutine vorhanden ist, fordert der Urlader den Anwender auf, eine Systemreferenzdiskette in ein Diskettenlaufwerk einzulegen und die "Y"-Taste auf der Tastatur zu drücken, Schritt 617. Dann wartete der Urlader, bis die Taste gedrückt

wird, Schritt 618. Sobald die Taste gedrückt wird, überprüft der Urlader, ob eine gültige Referenzdiskette vorliegt, Schritt 619. Wenn nicht, wiederholt der Urlader den Prozeß ab Schritt 617.

Wieder zurück zu Schritt 619, wenn der Urlader eine gültige Systemreferenzdiskette findet, weist er das BIOS an, das Schutzmittel für die Systempartition zu deaktivieren und die Systempartition als Ursprung der Bootfestplatte einzurichten durch Anwenden des in Schritt 506 als logische Startblockadresse errechneten Werts, Schritte 640 und 650. Die Systempartition ist jetzt ungeschützt. Der Urlader holt die Startroutine von der Referenzdiskette und gibt die Steuerung an sie, Schritt 660. Die Startroutine bootet die Systemreferenzdiskette.

Fig. 13 zeigt die BIOS-Modifikation, die erforderlich ist, das Booten des Systemreferenz-Diskettenbilds von der Systempartition der Boot-Festplatte zu unterstützen, oder um den Zugriff auf das Bild zu ermöglichen, wenn eine Systemreferenzdiskette gebootet wird. Wenn das BIOS eine Anforderung erhält, eine Datenübertragungsoperation vorzunehmen, bestimmt es, ob das diese Boot-Festplatte ist, wie in Schritt 700 gezeigt wird. Die Boot-Festplatte ist die erste physikalische Festplatte im Festplattenadapter. Wenn die Festplatte nicht die Boot-Festplatte ist, führt das BIOS die gewünschte Operation durch, Schritt 730.

Wieder zurück zu Schritt 700; wenn die Festplatte die Boot-Festplatte ist, prüft BIOS, ob das Systempartitions-Flag Wahr ist oder eine Systemreferenzdiskette gebootet wird; Schritt 710. Wenn keines der beiden richtig ist, führt BIOS die geforderte Operation durch, Schritt 730.

Wieder zurück zu Schritt 710; wenn das Systempartition-Boot-Flag Wahr ist oder eine Systemreferenzdiskette gebootet wird, wird die in Schritt 506 berechnete Festplatten-Blockadresse zu einer beliebigen Blockadresse hinzugefügt, nach dem Umwandeln von den vom Anwender beigestellten Zylinder-, Kopf- und Sektorparametern, versehen mit der Anforderung einer Festplattendaten-Übertragungsfunktion, Schritt 720. Dadurch sieht die Systempartition aus, als ob sie der erste Block auf der Festplatte sei. Somit sieht die Systempartition so aus, als ob sie die aktive Partition auf der Boot-Festplatte sei. Dann führt das BIOS die angeforderte Operation aus, Schritt 730.

Hier wurde ein Verfahren und ein Gerät zum Urladen des Systemreferenzdiskettenbilds von der Systempartition aus einer Massenspeichervorrichtung gezeigt, wie z.B. aus einem Festplattenlaufwerk. Die Systempartition ist vorgesehen durch Schützen eines Bereich auf dem Diskettenlaufwerk. Die Systempartition wird bootfähig gemacht durch Speichern der Startadresse der Systempartition auf dem Plattenlaufwerk und Anweisung an den BIOS, diese als den Ursprung der Festplatte zu benutzen, wenn ein Urladen des Systembezugsdiskettenbildes angefordert oder nötig ist. Durch Vorsehen dieser Fähigkeit sind die Systemreferenzdisketten-Dienstprogramme automatisch jederzeit verfügbar, wenn die Konfiguration verändert wird, ein System-Dienstprogramm gewünscht wird, oder während der Ausführung des POST ein Fehler entdeckt wurde. Das verstärkt die Anwendbarkeit des Systems.

A N S P R Ü C H E

1. Ein Personalcomputersystem mit einem Systemprozessor (26) zum Abarbeiten eines Betriebssystems, einem Festwertspeicher (36), einem Speicher mit wahlfreiem Zugriff (32) und Peripheriegeräten einschließlich wenigstens einer Direktzugriffsspeichervorrichtung (62, 66), wobei das System ein Schnittstellenprogramm zur Steuerung der Datenbewegung in den bzw. aus dem Prozessor aufweist, dadurch gekennzeichnet, daß die Direktzugriffsspeichervorrichtung ein Schutzmittel aufweist zum Schützen eines Bereichs der Direktzugriffsspeichervorrichtung, in der ein Teil des Schnittstellenprogramms gespeichert ist, das Schutzmittel den Zugriff auf den geschützten Bereich als Reaktion auf ein Rückstellsignal zuläßt, dieser Teil des Schnittstellenprogramms betriebsfähig ist, wenn er in den Speicher mit wahlfreiem Zugriff geladen ist, um das Betriebssystem zu urladen und das Schutzmittel zu aktivieren, um den Zugriff auf den Schutzbereich der Direktzugriffsspeichervorrichtung während des Abarbeitens des Betriebssystems zu verhindern, wobei der Schutzbereich der Direktzugriffsspeichervorrichtung ferner Systemdienstprogramme beinhaltet, die abgearbeitet werden, sobald beim Laden des Betriebsystems ein Fehlerzustand erkannt wird.

2. Ein System gemäß Anspruch 1, in dem die Direktzugriffsspeichervorrichtung eine Festplatte enthält.

3. Ein System gemäß Anspruch 1 oder 2, in dem die Systemdienstprogramme ein Programm zur Modifizierung der Systemkonfiguration aufweisen.

4. Ein System gemäß Anspruch 1, 2 oder 3, in dem ein Initialisierungsteil des Schnittstellenprogramms im Festwertspeicher abgespeichert ist und betrieben werden kann, um den Systemprozessor zu initialisieren und die Generierung eines Rückstellsignals für die Direktzugriffsspeichervorrichtung anlaufen zu lassen, um den Zugriff darauf zuzulassen.

10

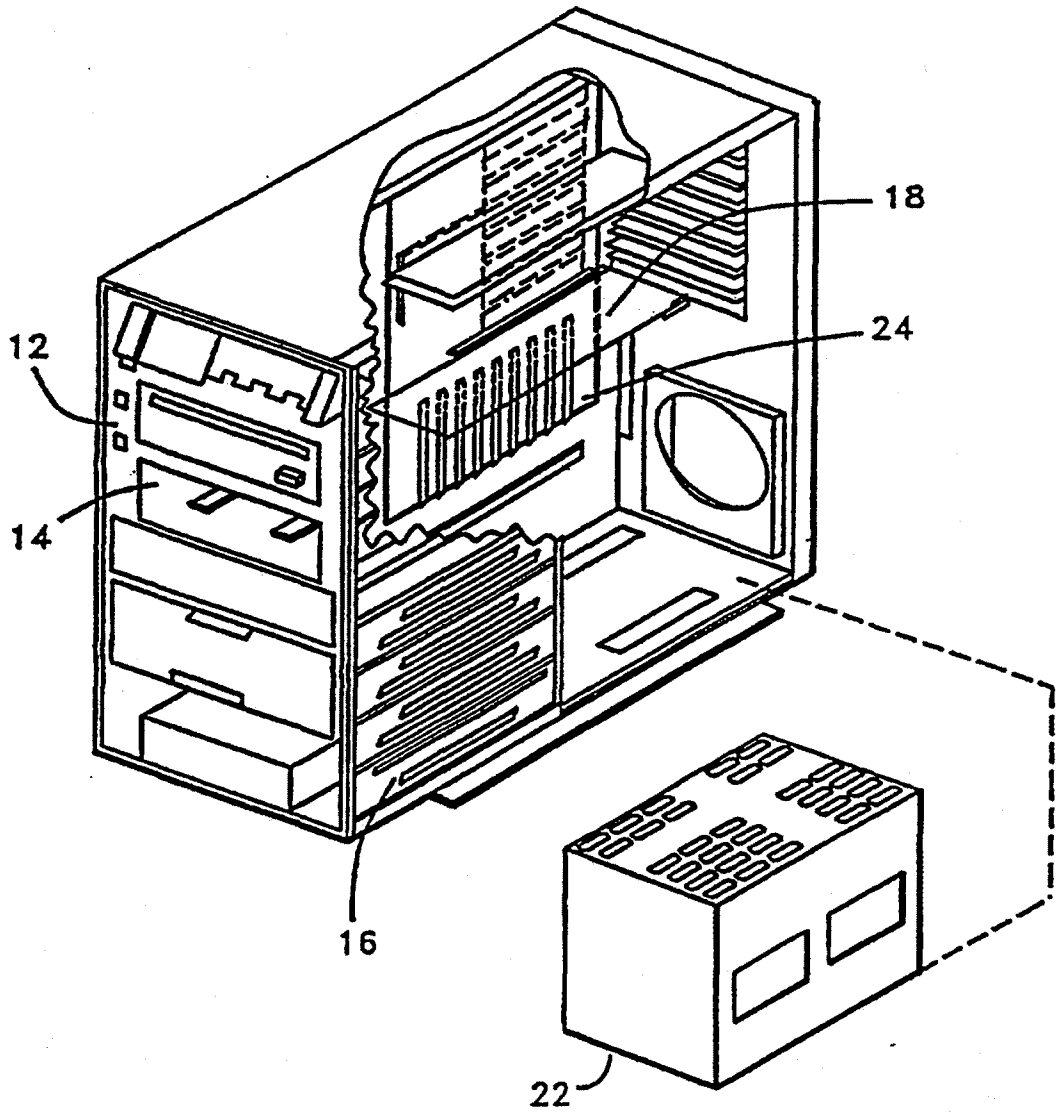

FIG 1

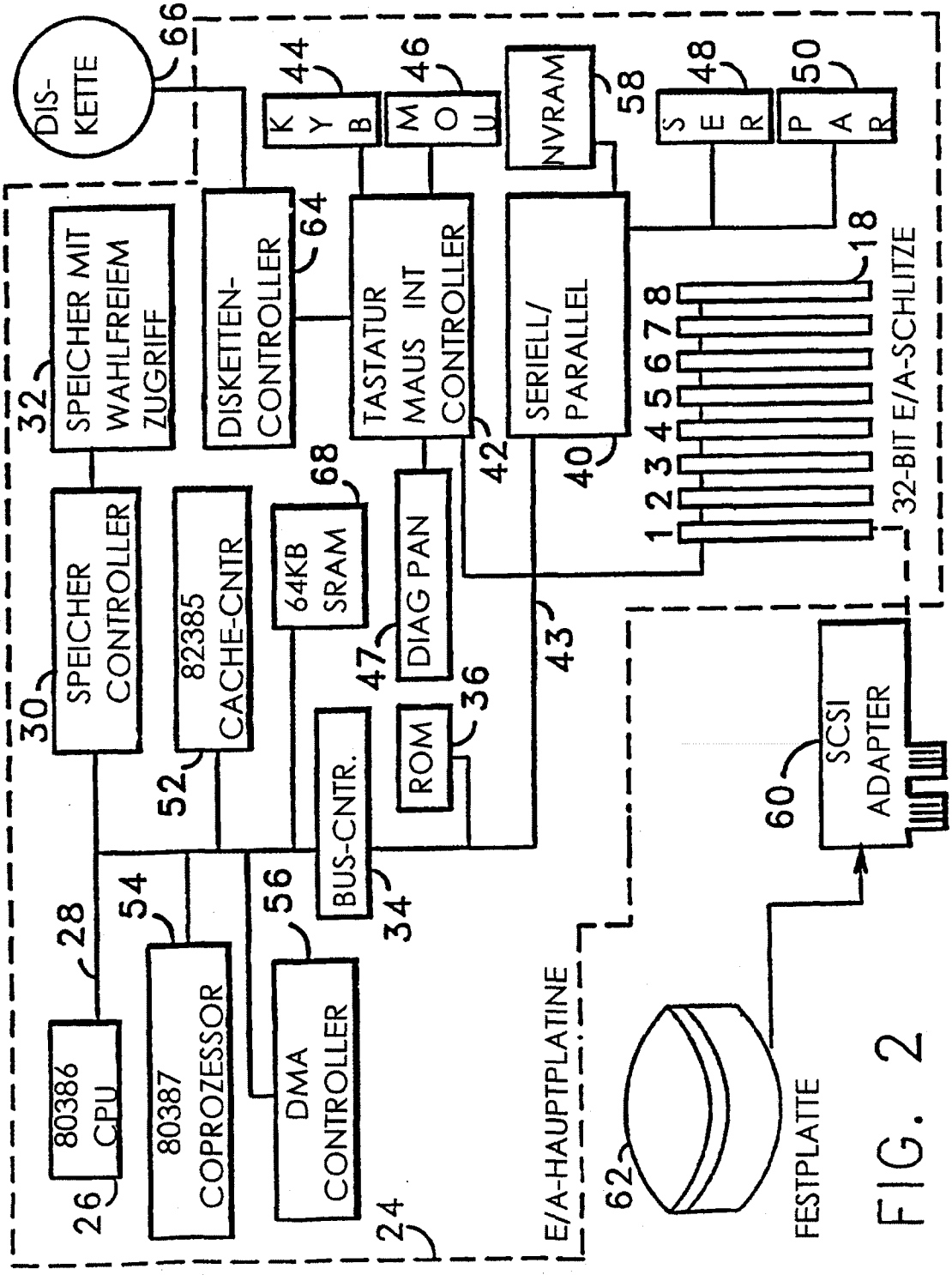
DIS-KETTE
66
K Y B
44
M O U
46
NVRAM
58
S E R
48
P A R
50
SPEICHER MIT WAHLFREIEM ZUGRIFF
32
DISKETTEN-CONTROLLER
64
TASTATUR MAUS INT CONTROLLER
42
SERIELL/ PARALLEL
40
18
1 2 3 4 5 6 7 8
32-BIT E/A-SCHLITZE
SPEICHER CONTROLLER
30
82385 CACHE-CNTR
52
64KB SRAM
68
DIAG PAN
47
43
ROM
36
BUS-CNTR.
34
SCSI ADAPTER
60
80386 CPU
26
28
80387 COPROZESSOR
54
DMA CONTROLLER
56
24
E/A-HAUPTPLATINE
62
FESTPLATTE

FIG. 2

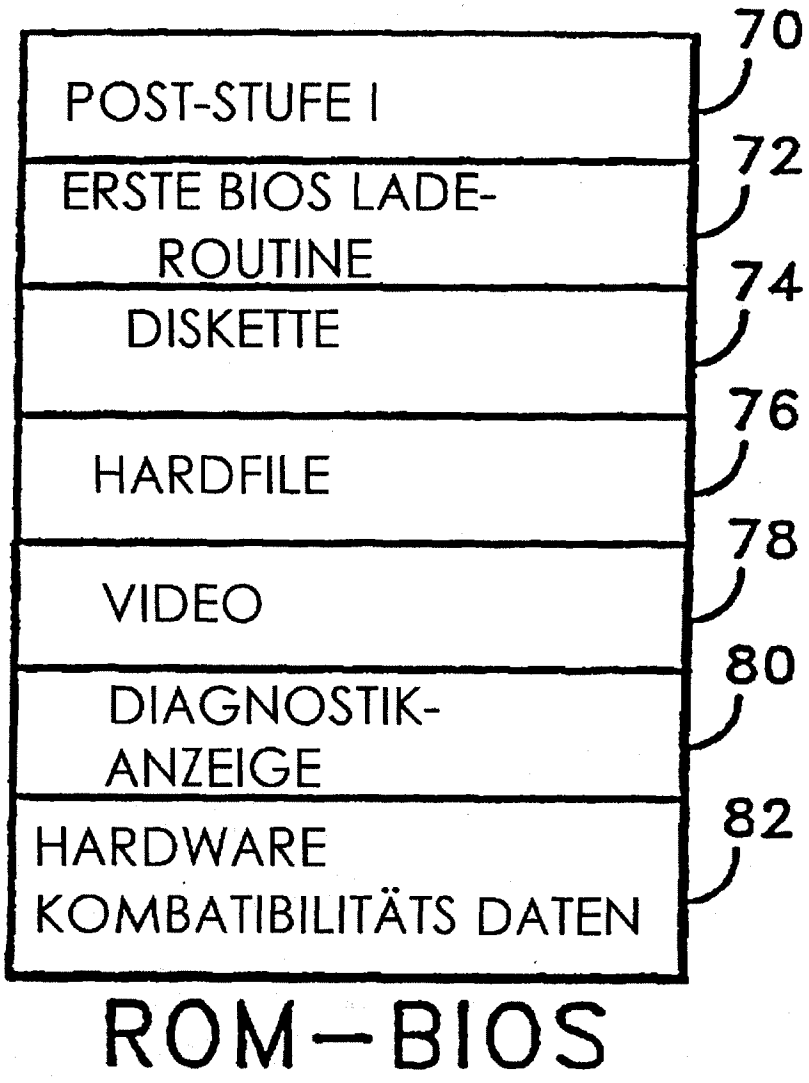
POST-STUFE I
70
ERSTE BIOS LADE-ROUTINE
72
DISKETTE
74
HARDFILE
76
VIDEO
78
DIAGNOSTIK-ANZEIGE
80
HARDWARE KOMBATIBILITÄTS DATEN
82
ROM-BIOS

FIG. 3

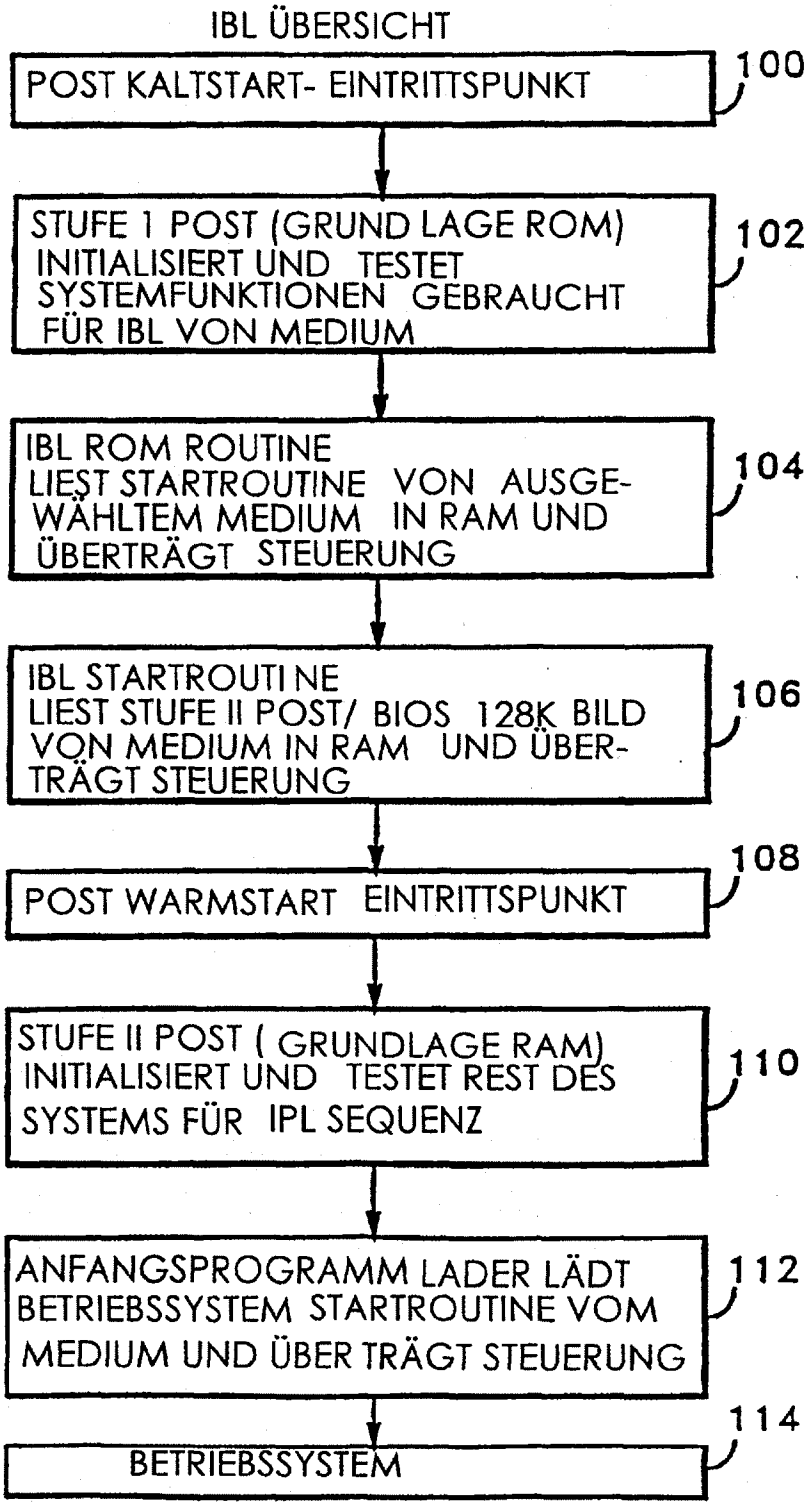
IBL ÜBERSICHT
POST KALTSTART- EINTRITTSPUNKT
100
STUFE 1 POST (GRUND LAGE ROM) INITIALISIERT UND TESTET SYSTEMFUNKTIONEN GEBRAUCHT FÜR IBL VON MEDIUM
102
IBL ROM ROUTINE LIEST STARTROUTINE VON AUSGE-WÄHLTEM MEDIUM IN RAM UND ÜBERTRÄGT STEUERUNG
104
IBL STARTROUTI NE LIEST STUFE II POST/ BIOS 128K BILD VON MEDIUM IN RAM UND ÜBER-TRÄGT STEUERUNG
106
POST WARMSTART EINTRITTSPUNKT
108
STUFE II POST (GRUNDLAGE RAM) INITIALISIERT UND TESTET REST DES SYSTEMS FÜR IPL SEQUENZ
110
ANFANGSPROGRAMM LADER LÄDT BETRIEBSSYSTEM STARTROUTINE VOM MEDIUM UND ÜBER TRÄGT STEUERUNG
112
BETRIEBSSYSTEM
114

FIG. 4

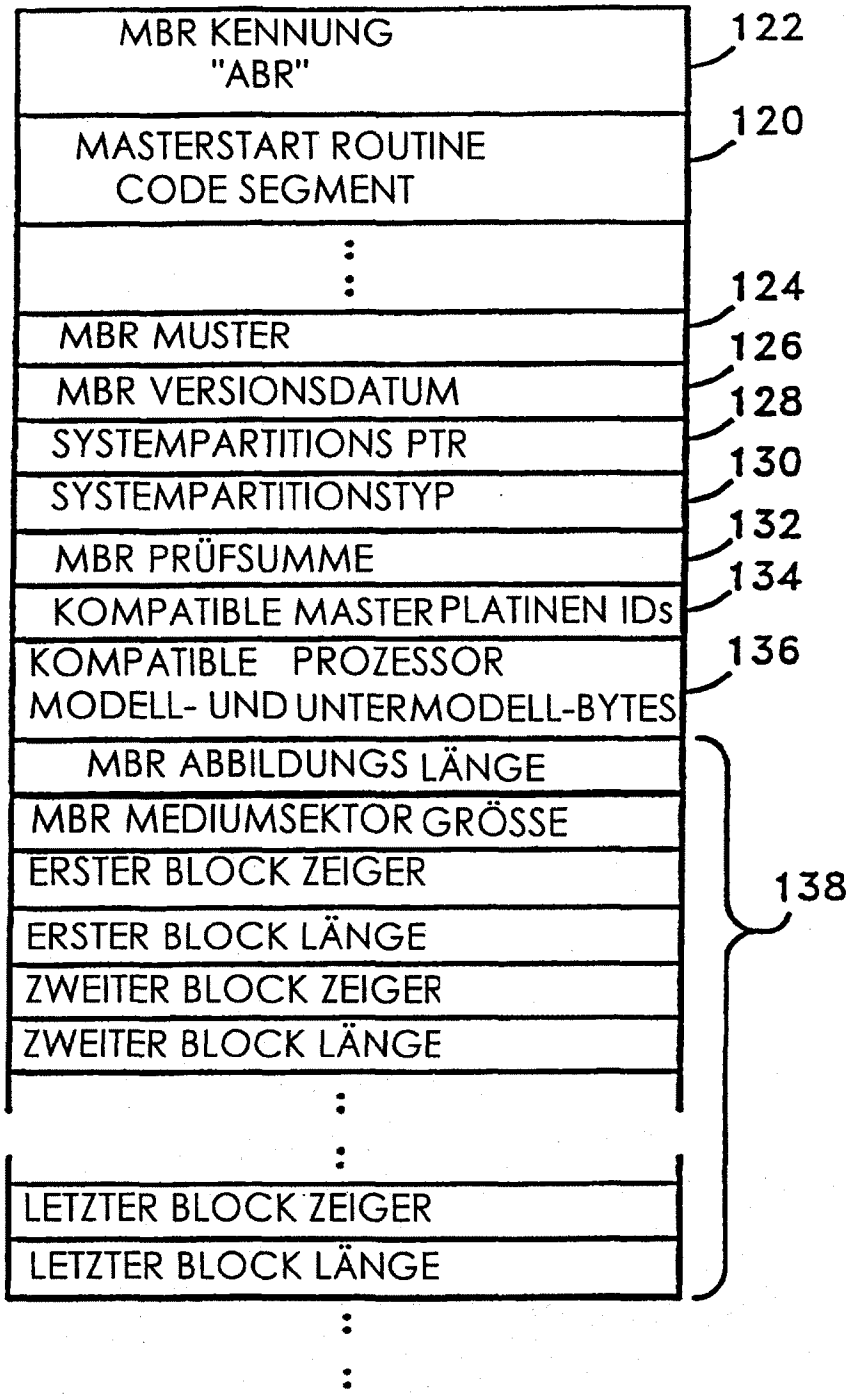
MBR KENNUNG "ABR"
122
MASTERSTART ROUTINE CODE SEGMENT
120
124
MBR MUSTER
126
MBR VERSIONSDATUM
128
SYSTEMPARTITIONS PTR
130
SYSTEMPARTITIONSTYP
132
MBR PRÜFSUMME
134
KOMPATIBLE MASTER PLATINEN IDs
136
KOMPATIBLE PROZESSOR MODELL- UND UNTERMODELL-BYTES
MBR ABBILDUNGS LÄNGE
MBR MEDIUMSEKTOR GRÖSSE
ERSTER BLOCK ZEIGER
ERSTER BLOCK LÄNGE
ZWEITER BLOCK ZEIGER
ZWEITER BLOCK LÄNGE
LETZTER BLOCK ZEIGER
LETZTER BLOCK LÄNGE
138

FIG. 5

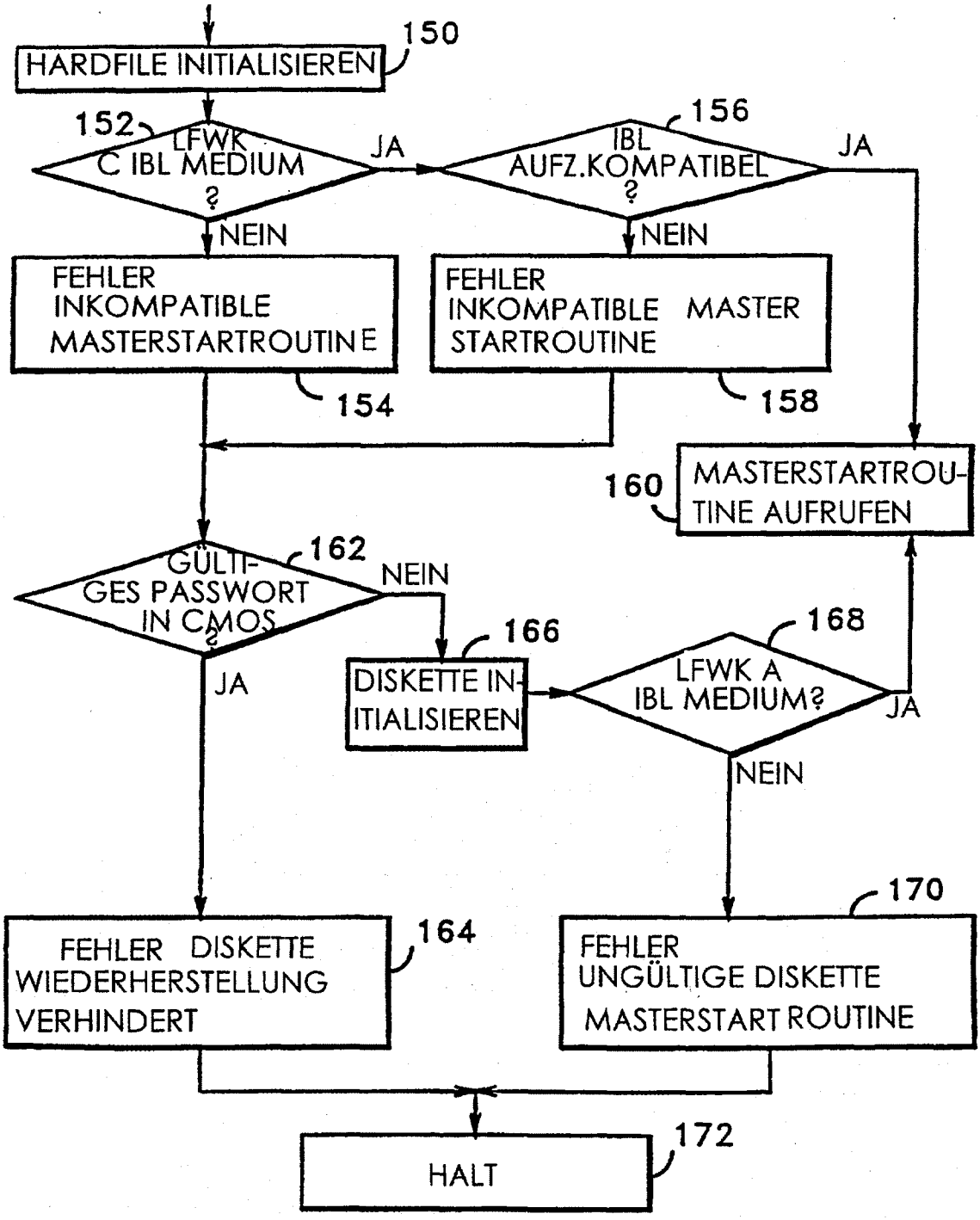
HARDFILE INITIALISIEREN
150
152
LFWK C IBL MEDIUM ?
JA
NEIN
156
IBL AUFZ.KOMPATIBEL ?
JA
NEIN
FEHLER INKOMPATIBLE MASTERSTARTROUTINE
154
FEHLER INKOMPATIBLE MASTER STARTROUTINE
158
160
MASTERSTARTROUTINE AUFRUFEN
162
GÜLTIGES PASSWORT IN CMOS ?
NEIN
JA
166
DISKETTE INITIALISIEREN
168
LFWK A IBL MEDIUM?
JA
NEIN
FEHLER DISKETTE WIEDERHERSTELLUNG VERHINDERT
164
170
FEHLER UNGÜLTIGE DISKETTE MASTERSTART ROUTINE
HALT
172

FIG. 6A

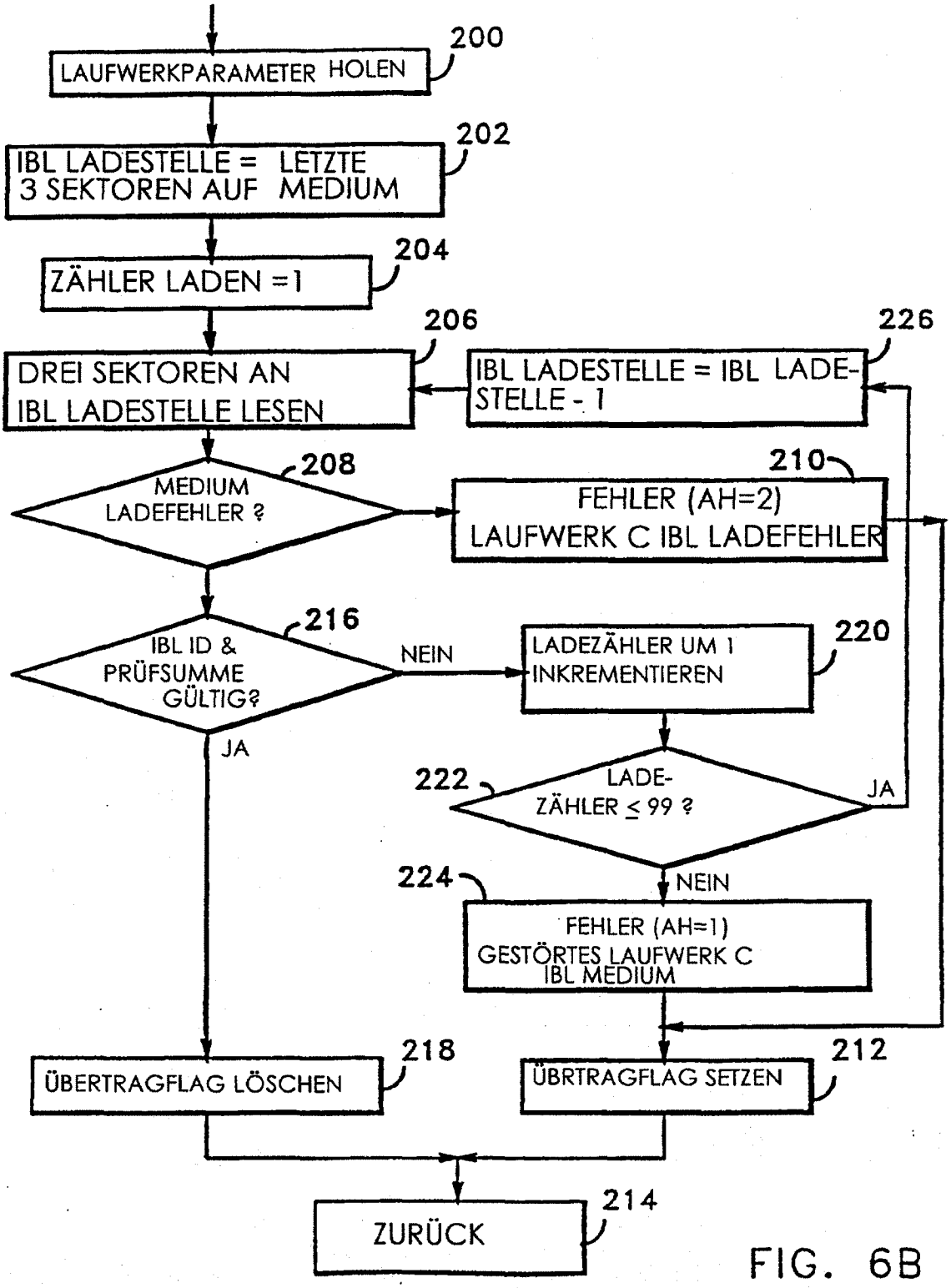
LAUFWERKPARAMETER HOLEN
200
IBL LADESTELLE = LETZTE 3 SEKTOREN AUF MEDIUM
202
ZÄHLER LADEN =1
204
DREI SEKTOREN AN IBL LADESTELLE LESEN
206
IBL LADESTELLE = IBL LADE-STELLE - 1
226
MEDIUM LADEFEHLER ?
208
FEHLER (AH=2) LAUFWERK C IBL LADEFEHLER
210
IBL ID & PRÜFSUMME GÜLTIG?
216
NEIN
LADEZÄHLER UM 1 INKREMENTIEREN
220
JA
LADE-ZÄHLER ≤ 99 ?
222
JA
NEIN
224
FEHLER (AH=1) GESTÖRTES LAUFWERK C IBL MEDIUM
ÜBERTRAGFLAG LÖSCHEN
218
ÜBRTRAGFLAG SETZEN
212
ZURÜCK
214

FIG. 6B

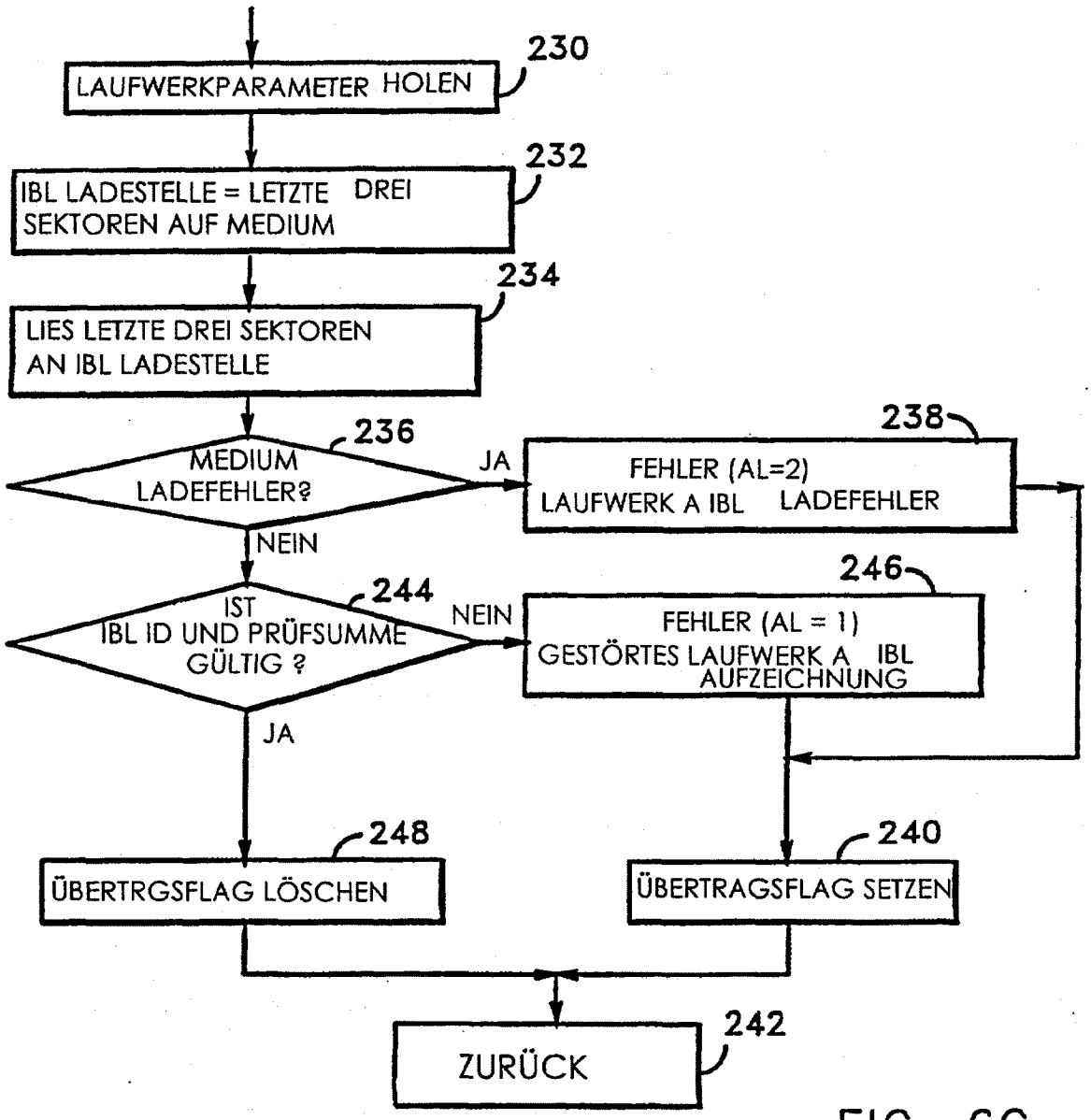
LAUFWERKPARAMETER HOLEN
230
IBL LADESTELLE = LETZTE DREI SEKTOREN AUF MEDIUM
232
LIES LETZTE DREI SEKTOREN AN IBL LADESTELLE
234
MEDIUM LADEFEHLER?
236
JA
FEHLER (AL=2) LAUFWERK A IBL LADEFEHLER
238
NEIN
IST IBL ID UND PRÜFSUMME GÜLTIG ?
244
NEIN
FEHLER (AL = 1) GESTÖRTES LAUFWERK A IBL AUFZEICHNUNG
246
JA
ÜBERTRGSFLAG LÖSCHEN
248
ÜBERTRAGSFLAG SETZEN
240
ZURÜCK
242

FIG. 6C

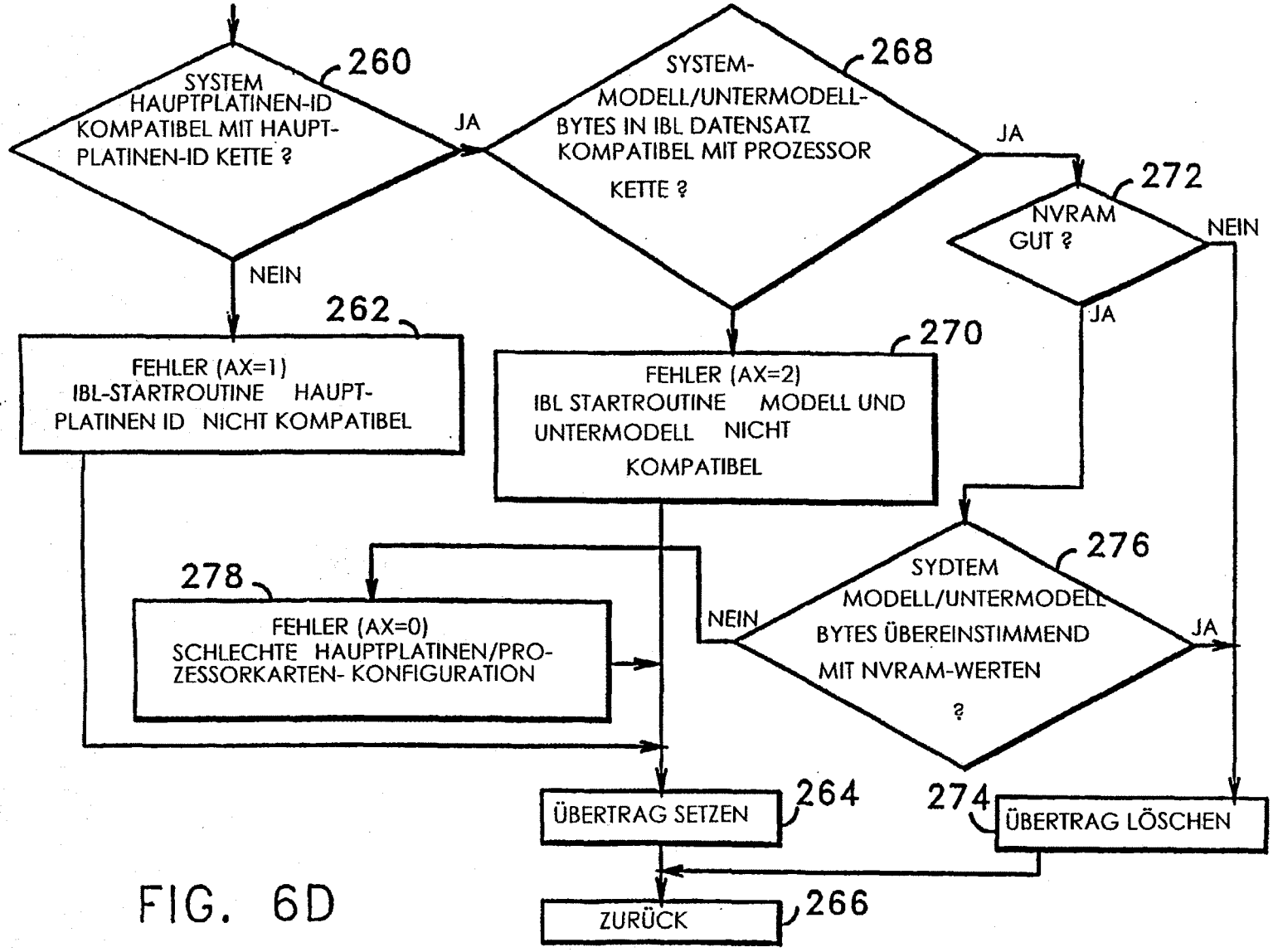
260
SYSTEM HAUPTPLATINEN-ID KOMPATIBEL MIT HAUPT-PLATINEN-ID KETTE ?
JA
NEIN
262
FEHLER (AX=1) IBL-STARTROUTINE HAUPT-PLATINEN ID NICHT KOMPATIBEL
268
SYSTEM-MODELL/UNTERMODELL-BYTES IN IBL DATENSATZ KOMPATIBEL MIT PROZESSOR KETTE ?
JA
270
FEHLER (AX=2) IBL STARTROUTINE MODELL UND UNTERMODELL NICHT KOMPATIBEL
272
NVRAM GUT ?
NEIN
JA
276
SYDTEM MODELL/UNTERMODELL BYTES ÜBEREINSTIMMEND MIT NVRAM-WERTEN ?
NEIN
JA
278
FEHLER (AX=0) SCHLECHTE HAUPTPLATINEN/PRO-ZESSORKARTEN- KONFIGURATION
ÜBERTRAG SETZEN
264
274
ÜBERTRAG LÖSCHEN
ZURÜCK
266

FIG. 6D

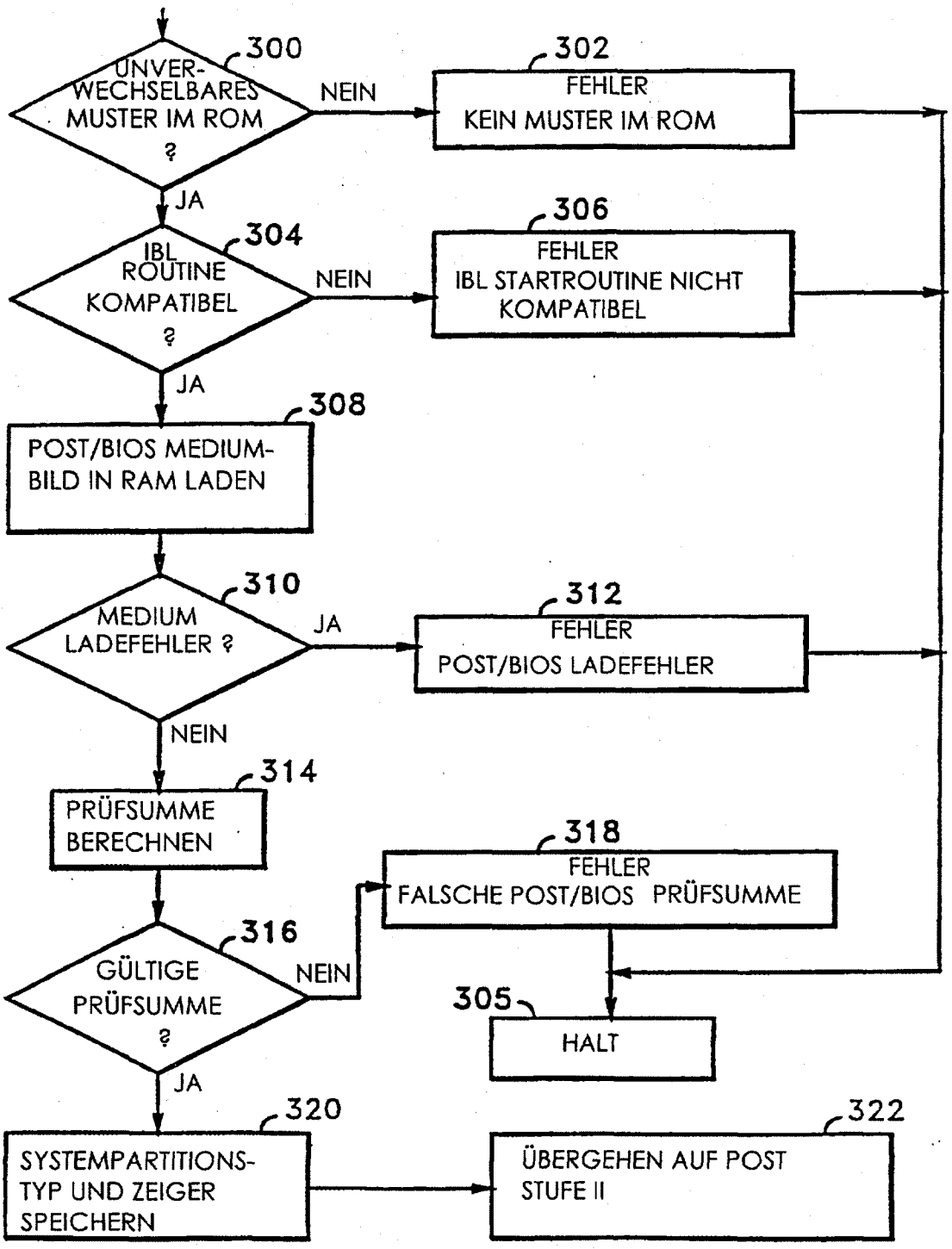
300
UNVER-
WECHSELBARES
MUSTER IM ROM
?
NEIN
302
FEHLER
KEIN MUSTER IM ROM
JA
304
IBL
ROUTINE
KOMPATIBEL
?
NEIN
306
FEHLER
IBL STARTROUTINE NICHT
KOMPATIBEL
JA
308
POST/BIOS MEDIUM-
BILD IN RAM LADEN
310
MEDIUM
LADEFEHLER ?
JA
312
FEHLER
POST/BIOS LADEFEHLER
NEIN
314
PRÜFSUMME
BERECHNEN
318
FEHLER
FALSCHE POST/BIOS PRÜFSUMME
316
GÜLTIGE
PRÜFSUMME
?
NEIN
305
HALT
JA
320
SYSTEMPARTITIONS-
TYP UND ZEIGER
SPEICHERN
322
ÜBERGEHEN AUF POST
STUFE II

FIG. 7

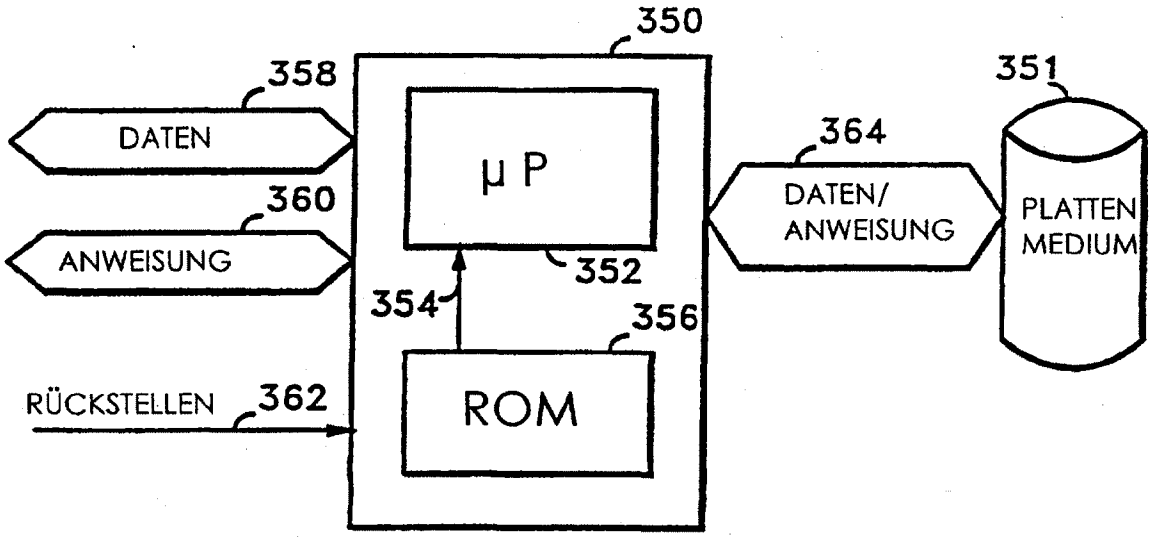
350
358
DATEN
360
ANWEISUNG
μ P
352
354
356
ROM
RÜCKSTELLEN
362
364
DATEN/
ANWEISUNG
351
PLATTEN
MEDIUM

FIG. 8

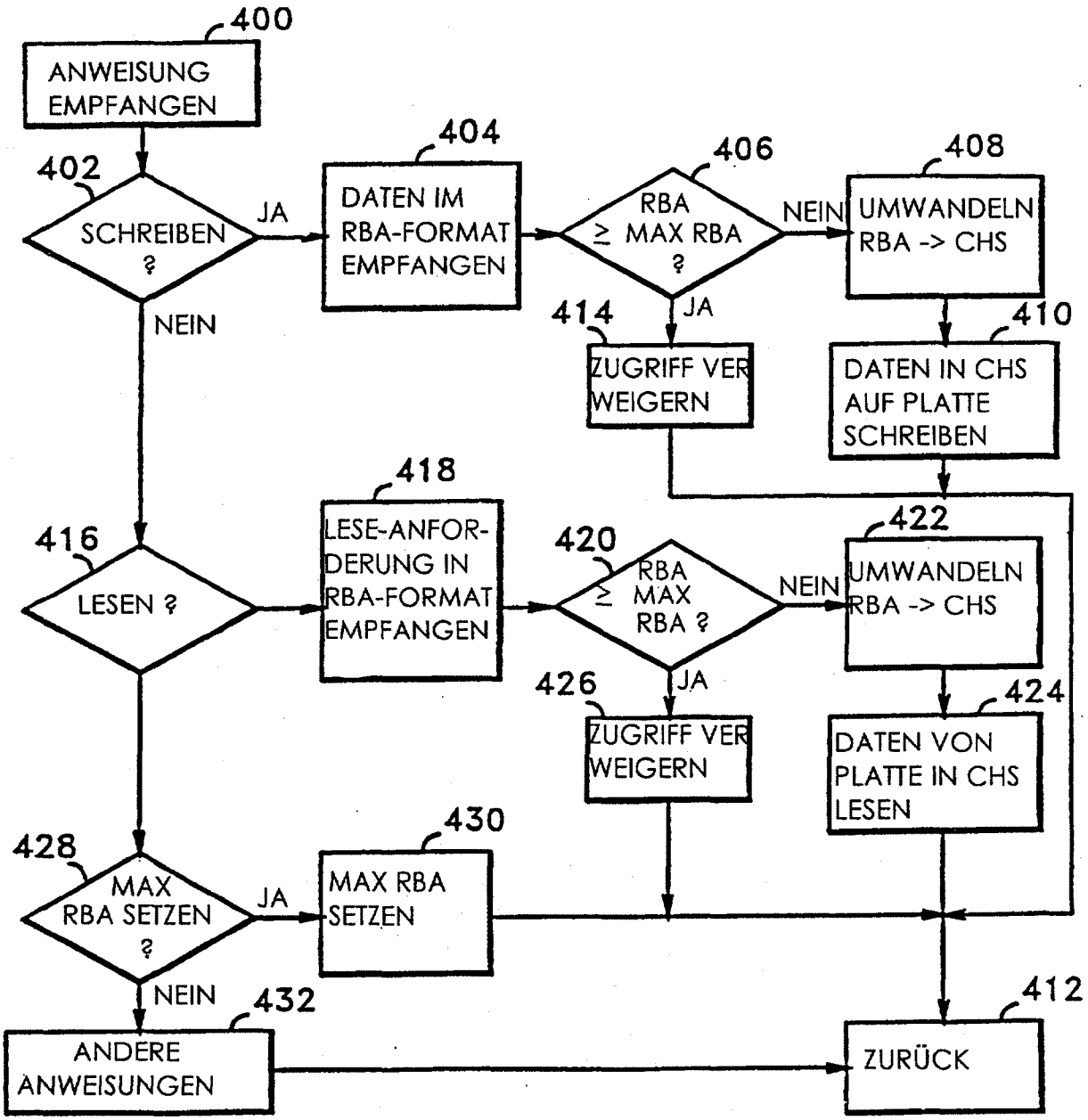
400
ANWEISUNG EMPFANGEN
402
SCHREIBEN ?
JA
NEIN
404
DATEN IM RBA-FORMAT EMPFANGEN
406
RBA ≥ MAX RBA ?
NEIN
JA
408
UMWANDELN RBA -> CHS
410
DATEN IN CHS AUF PLATTE SCHREIBEN
414
ZUGRIFF VERWEIGERN
416
LESEN ?
418
LESE-ANFORDERUNG IN RBA-FORMAT EMPFANGEN
420
RBA ≥ MAX RBA ?
NEIN
JA
422
UMWANDELN RBA -> CHS
424
DATEN VON PLATTE IN CHS LESEN
426
ZUGRIFF VERWEIGERN
428
MAX RBA SETZEN ?
JA
NEIN
430
MAX RBA SETZEN
432
ANDERE ANWEISUNGEN
412
ZURÜCK

FIG 9

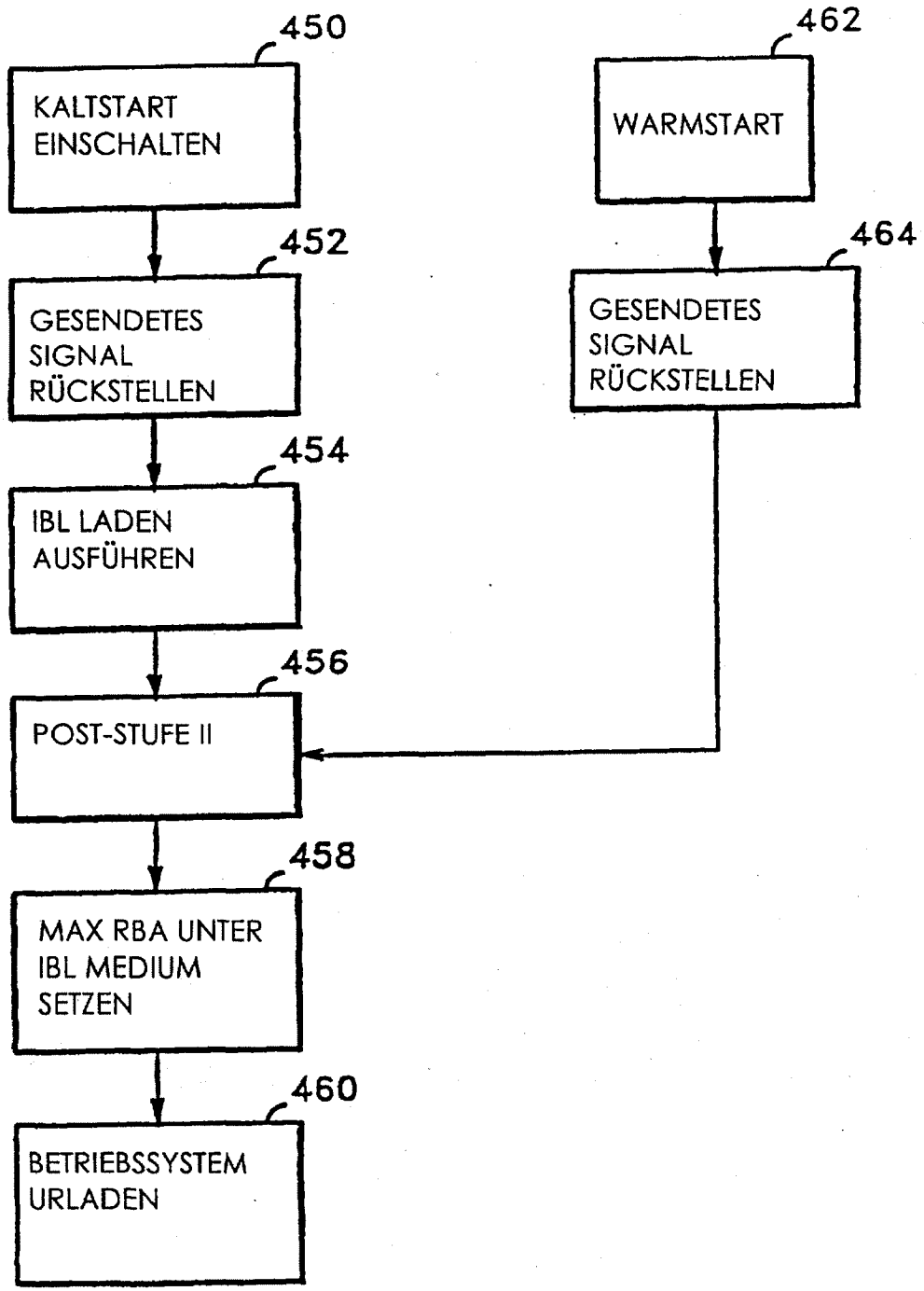
450
KALTSTART EINSCHALTEN
452
GESENDETES SIGNAL RÜCKSTELLEN
454
IBL LADEN AUSFÜHREN
456
POST-STUFE II
458
MAX RBA UNTER IBL MEDIUM SETZEN
460
BETRIEBSSYSTEM URLADEN
462
WARMSTART
464
GESENDETES SIGNAL RÜCKSTELLEN

FIG. 10

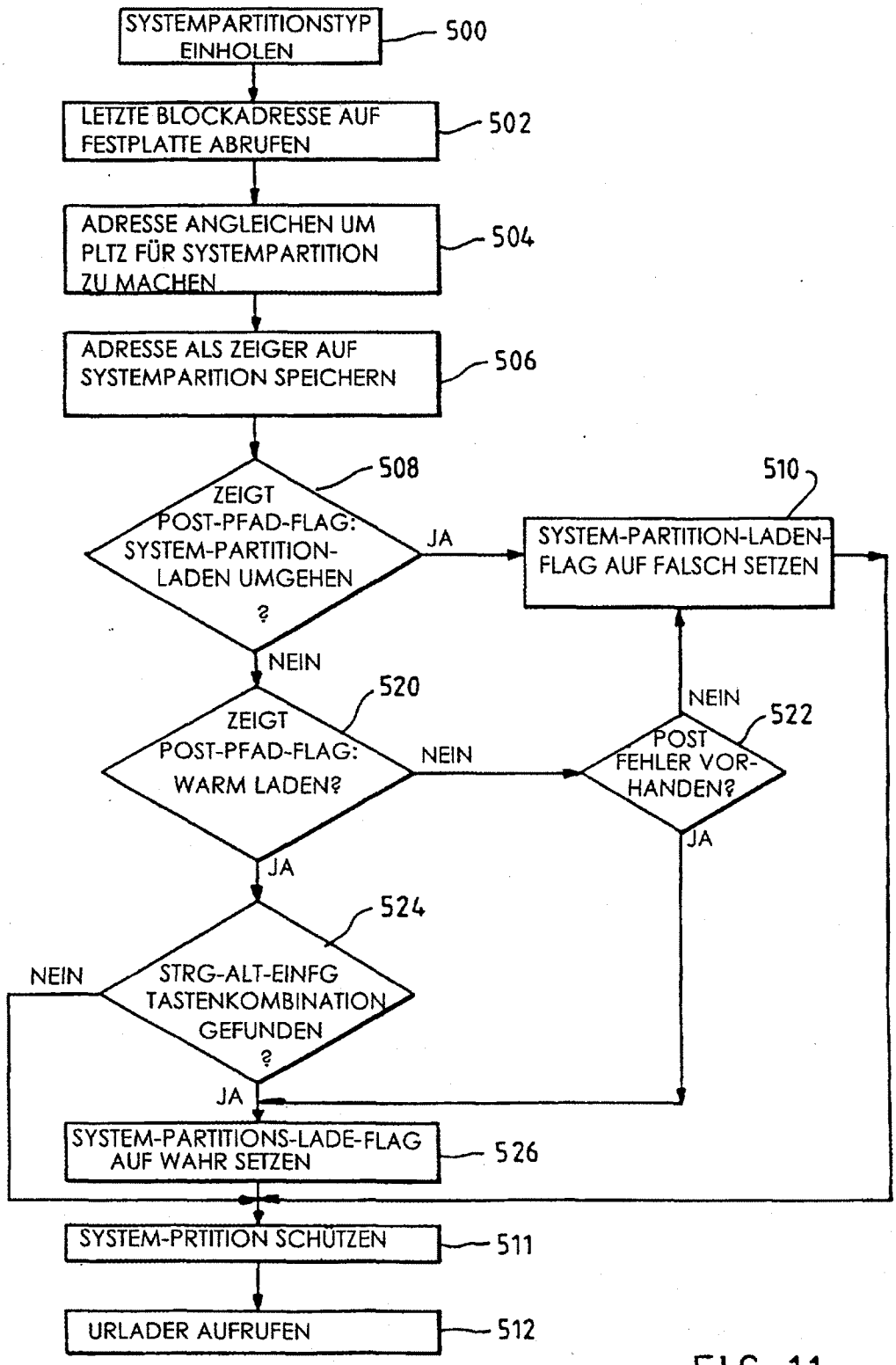
SYSTEMPARTITIONSTYP EINHOLEN
500
LETZTE BLOCKADRESSE AUF FESTPLATTE ABRUFEN
502
ADRESSE ANGLEICHEN UM PLTZ FÜR SYSTEMPARTITION ZU MACHEN
504
ADRESSE ALS ZEIGER AUF SYSTEMPARITION SPEICHERN
506
ZEIGT POST-PFAD-FLAG: SYSTEM-PARTITION-LADEN UMGEHEN ?
508
JA
SYSTEM-PARTITION-LADEN-FLAG AUF FALSCH SETZEN
510
NEIN
ZEIGT POST-PFAD-FLAG: WARM LADEN?
520
NEIN
POST FEHLER VORHANDEN?
522
NEIN
JA
JA
STRG-ALT-EINFG TASTENKOMBINATION GEFUNDEN ?
524
NEIN
JA
SYSTEM-PARTITIONS-LADE-FLAG AUF WAHR SETZEN
526
SYSTEM-PRTITION SCHÜTZEN
511
URLADER AUFRUFEN
512

FIG. 11

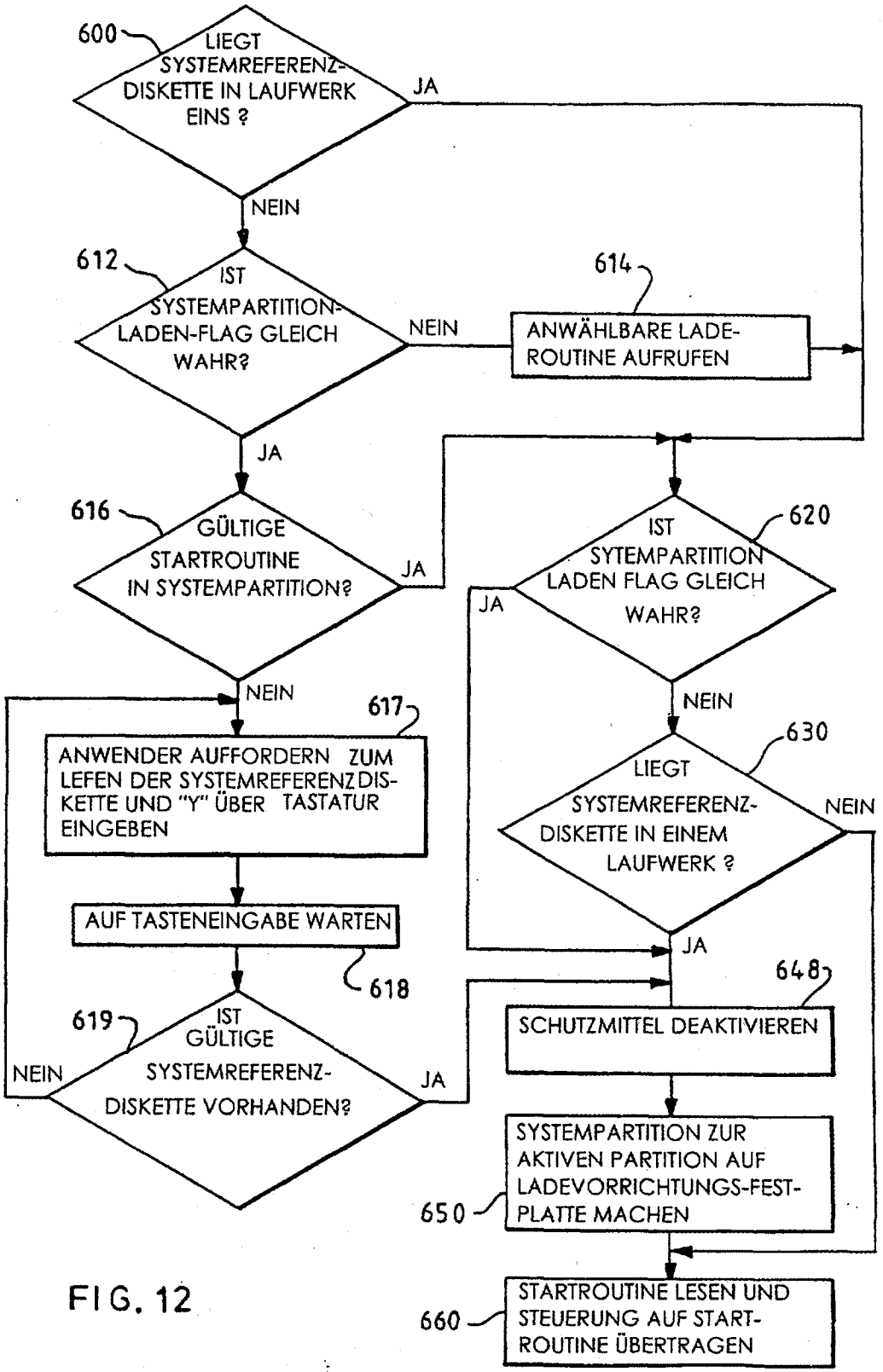
600
LIEGT SYSTEMREFERENZ-DISKETTE IN LAUFWERK EINS ?
JA
NEIN
612
IST SYSTEMPARTITION-LADEN-FLAG GLEICH WAHR?
NEIN
614
ANWÄHLBARE LADE-ROUTINE AUFRUFEN
JA
616
GÜLTIGE STARTROUTINE IN SYSTEMPARTITION?
JA
NEIN
617
ANWENDER AUFFORDERN ZUM LEFEN DER SYSTEMREFERENZDISKETTE UND "Y" ÜBER TASTATUR EINGEBEN
AUF TASTENEINGABE WARTEN
618
619
IST GÜLTIGE SYSTEMREFERENZ-DISKETTE VORHANDEN?
NEIN
JA
620
IST SYTEMPARTITION LADEN FLAG GLEICH WAHR?
JA
NEIN
630
LIEGT SYSTEMREFERENZ-DISKETTE IN EINEM LAUFWERK ?
NEIN
JA
648
SCHUTZMITTEL DEAKTIVIEREN
SYSTEMPARTITION ZUR AKTIVEN PARTITION AUF LADEVORRICHTUNGS-FEST-PLATTE MACHEN
650
STARTROUTINE LESEN UND STEUERUNG AUF START-ROUTINE ÜBERTRAGEN
660

FIG. 12

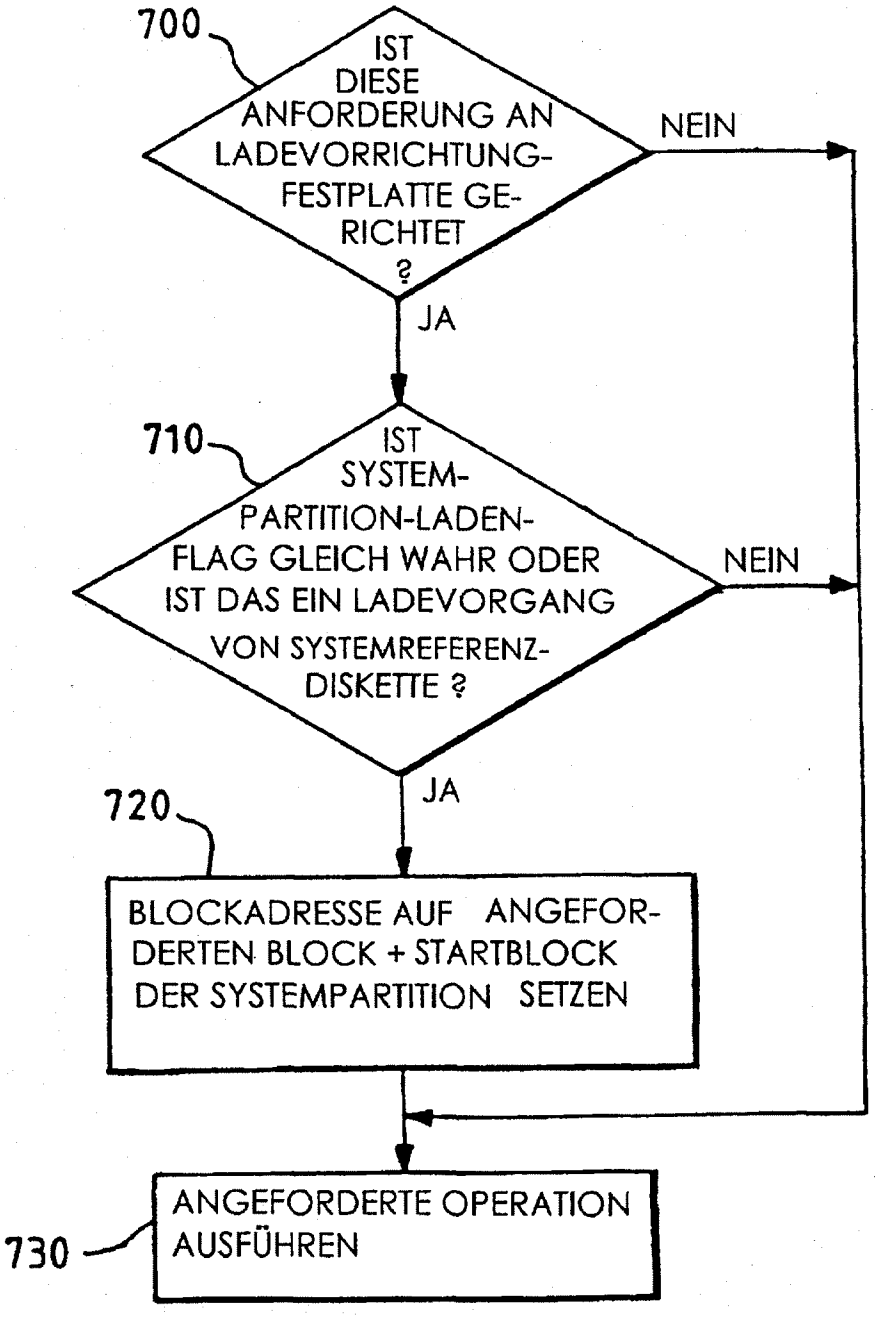
700
IST DIESE ANFORDERUNG AN LADEVORRICHTUNG-FESTPLATTE GE-RICHTET ?
NEIN
JA
710
IST SYSTEM-PARTITION-LADEN-FLAG GLEICH WAHR ODER IST DAS EIN LADEVORGANG VON SYSTEMREFERENZ-DISKETTE ?
NEIN
JA
720
BLOCKADRESSE AUF ANGEFOR-DERTEN BLOCK + STARTBLOCK DER SYSTEMPARTITION SETZEN
730
ANGEFORDERTE OPERATION AUSFÜHREN

FIG. 13